

教育部高等学校电子信息类专业教学指导委员会规划教材
高等学校电子信息类专业系列教材

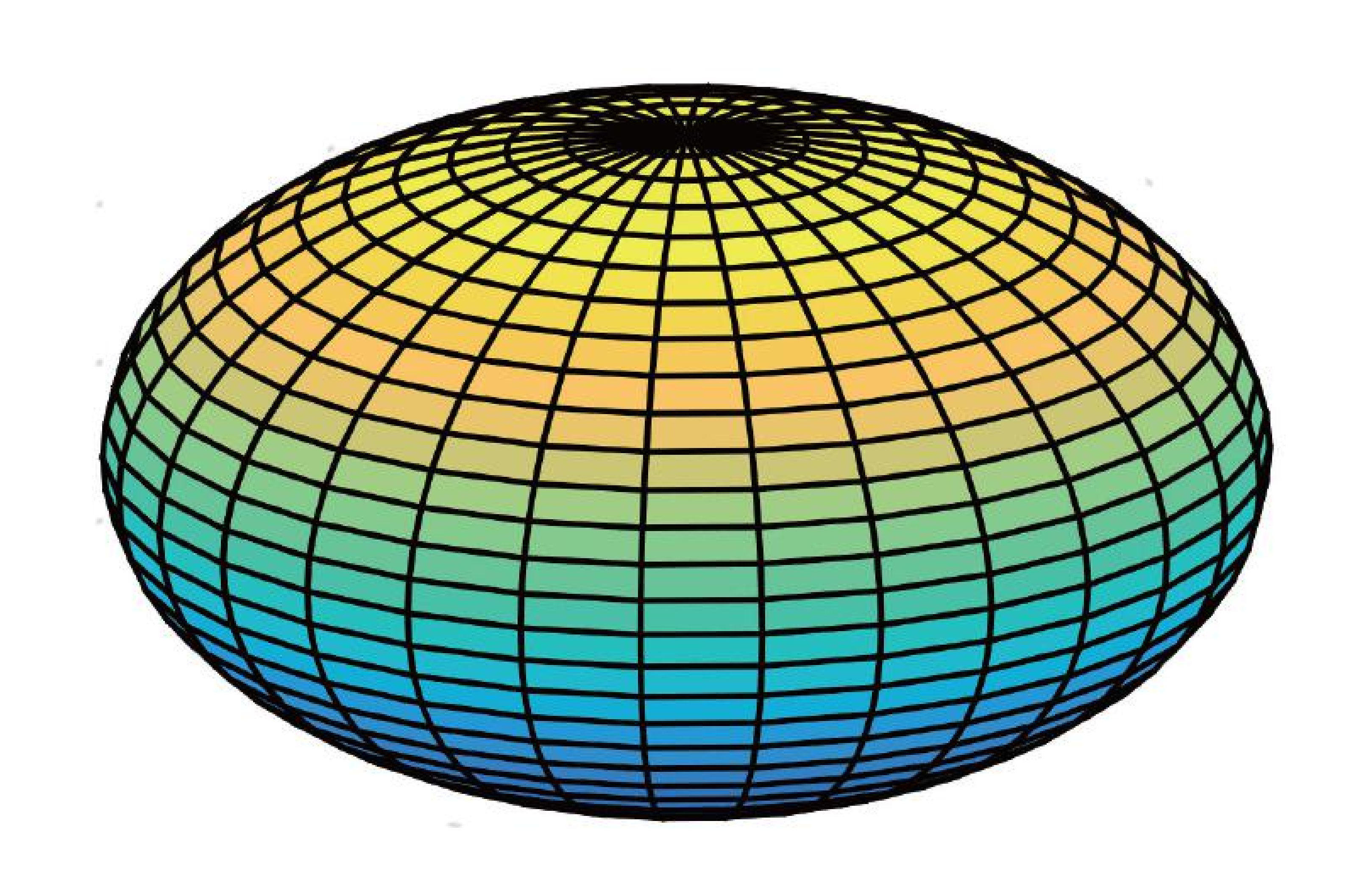




数字图像处理及应用

使用MATLAB分析与实现

陈天华　编著
Chen Tianhua



清华大学出版社

教育部高等学校电子信息类专业教学指导委员会规划教材
高等学校电子信息类专业系列教材

数字图像处理及应用

——使用MATLAB分析与实现

Digital Image Processing and Application: Using MATLAB

陈天华　编著
Chen Tianhua

清华大学出版社
北京

内容简介

本书系统地介绍了数字图像处理的基本概念和理论、基本方法和算法。全书共分12章，从内容上可分为3大部分。第1部分是数字图像处理的基础知识，包括图像处理的基本概念及相关基础。第2部分是数字图像处理方面的核心内容，系统地讨论了数字图像处理中的各种基本技术，包括图像处理的各种变换、图像处理的基本运算、图像空域增强技术、图像频域增强技术、彩色图像处理、图像复原和图像编码技术等。第3部分是图像分析和理解方面的内容，由浅入深地介绍了图像分割、数学形态学、图像特征与理解等方面的内容，为深入应用奠定坚实的基础。本书在内容安排上循序渐进、深入浅出，各章均配有丰富的例题和习题，力求突出重点、面向应用、提高能力、解决问题。

本书可作为高等院校电子信息工程、通信工程、信息与信号处理、电子科学与技术、信息工程、计算机科学与技术、软件工程、自动化、电气工程、生物医学工程、物联网、电视技术和遥感遥测等相关专业的高年级学生和研究生的图像处理教材，也可以作为工程技术人员或其他相关人员的参考书。

图书在版编目(CIP)数据

数字图像处理及应用：使用MATLAB分析与实现/陈天华编著. —北京：清华大学出版社，2019
(高等学校电子信息类专业系列教材)
ISBN 978-7-302-51504-3

Ⅰ. ①数…　Ⅱ. ①陈…　Ⅲ. ①数字图象处理－Matlab软件－高等学校－教材　Ⅳ. ①TN911.73

中国版本图书馆CIP数据核字(2018)第255180号

责任编辑：盛东亮
封面设计：李召霞
责任校对：梁　毅
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　**邮　　编**：100084
社 总 机：010-62770175　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn,010-62795954
印 装 者：三河市君旺印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　**印　　张**：22　**字　　数**：539千字
版　　次：2018年12月第1版　**印　　次**：2018年12月第1次印刷
定　　价：69.00元

产品编号：061102-01

前言

PREFACE

1974年，数码相机之父塞尚发明和设计了第一台基于CCD的数码原型机，从此，数字图像技术逐步进入了大众生活；20世纪60年代末，阿波罗登月计划实现了月球和地球之间数以万计的图像传输和处理，从此，图像处理技术引起了美国、中国、俄罗斯及欧洲一些科技大国的重视。特别是近20年来，数字图像处理技术得到快速发展，随着人工智能、大数据、物联网、数字地球、智慧城市、机器人技术及Internet的广泛应用，信息传输总量急剧增长。在各种信息中，图像信息以其信息量大和直观性、准确性、可靠性等一系列优点，成为人类获取信息的主要来源和利用信息的重要手段。

现代社会，图像处理技术已经应用和渗透到科学研究和日常生活的方方面面，并日益受到重视，在航天、军事、医学、科学研究等许多领域发挥着越来越重要的作用。在航空航天领域，图像处理技术已被用于高效、快速传输从宇宙空间拍摄各种图像，并进行快速处理，处理结果用于火箭发射监控、飞行控制、姿态调整、飞船返回等，或用于资源勘探、天体研究、地图测绘、气象预报、大气污染防控等研究。近年来，我国成功发射了嫦娥三号、天宫2号、天舟1号飞船，无论发射和回收，还是着陆地点选择、发射过程监测、飞行控制、舱内设备控制、故障检测和故障唤醒，图像处理技术始终发挥着不可替代的作用。在生物医学工程领域，图像处理的应用也非常广泛，它已用于各种射线照片、CT成像、超声影像、内窥镜像、核磁成像分析，帮助医生对疾病进行快速、准确诊断，挽回了无数人的生命。在军事领域，图像处理技术已成功用于卫星和遥感照片识别、无人机飞行与导航、高精度夜视瞄准设备等。数字图像处理技术的应用不胜枚举，基于图像处理技术开发出来的现代新型装备，其很多功能都是人工难以企及的。

实践的需要和成果的积淀促进了图像处理技术向更高水平发展，数字图像处理技术正是在这种应用的迫切需要和自身的不断发展之中得到迅速发展的学科。未来，图像处理技术的发展及应用与人类未来的生活联系之紧密、影响之深远是不可估量的。

本书共分12章，内容可以分为3大部分。第1部分(第1、2章)是数字图像处理的基础知识，主要内容包括图像处理学科概况、图像处理系统、研究内容、基本概念、图像类型、图像数字化方法、图像文件基本格式、图像质量评价、灰度直方图及图像的统计特征等。第2部分(第3～9章)是数字图像处理的核心内容，系统地讨论了数字图像处理中的各种基本理论、算法和应用，包括图像处理中的各种变换方法、图像基本运算、图像空域增强、图像频域增强、彩色图像处理、图像复原和图像编码等。第3部分(第10～12章)是图像分析方面的重要内容，由浅入深地介绍了图像分割与边缘检测、数学形态学、图像特征与理解方面的内容，为深入应用奠定坚实的基础。

本书是作者在多年从事图像处理开发和教学科研的基础上撰写而成的，力图体现以下

三个特点：第一，将图像处理理论和图像处理的应用紧密结合起来，理论严谨、概念清晰、结构合理，系统介绍数字图像处理技术的知识、内容和应用领域，使读者对数字图像处理学科有一个全面的了解；第二，紧密结合相关专业本科生和研究生的培养规划，介绍图像处理技术方面的新理论、新技术、新标准和新应用，使读者充分了解图像处理技术的新发展和新应用；第三，注重理论和实际的结合，各有关章节内容既具有一定的理论深度，也对典型的应用做了较详细的介绍，各章均安排了重要应用实例，为读者编程和进一步的应用研究打下扎实的基础。

本书可作为高等院校电子信息工程、通信工程、信息与信号处理、电子科学技术、信息工程、计算机科学与技术、软件工程、自动化、电气工程、生物医学工程、物联网、电视技术和遥感遥测等相关专业的高年级学生和研究生的图像处理教材，也可以作为工程技术人员或其他相关人员的参考书。为方便教学，本书同步更新了电子教案及教材的全部源程序，如有需要，可到清华大学出版社网站下载或与作者联系。

在本书的撰写过程中，西安电子科技大学姜建国教授提出了宝贵意见，清华大学信息工程学院电子信息系章毓晋教授提出了有益的建议，在 2017 年 11 月召开的教育部电子信息类专业教学指导委员会会议上还听取了很多专家的建议，在此表示衷心的感谢。

由于作者水平有限，书中缺点和疏漏之处在所难免，恳请读者批评指正。欢迎读者对本书提出宝贵意见和建议。作者的联系方式：cth188@sina.com，cth188@hotmail.com。

作　者

2019 年 1 月

目录

CONTENTS

第1章 数字图像处理概论

CHAPTER 1

从胶片相机的问世，到数码相机的普及，都标志着图像处理技术进入到了人们的日常生活之中，数字图像处理技术只有几十年的历史，是一门既年轻又酝酿着巨大潜力的学科，同时它又是一门充满活力的交叉学科，它与数学、光学、信息科学、工程技术、计算机技术、人工智能和航天技术等众多学科密切相关。21世纪人类已进入了数字智能时代，数字图像处理技术的重要性将越来越凸显，其应用领域将越来越广泛，从早期的军用和医学等个别领域发展到现代工业、农业、军事、地理信息、采矿、气象预报、医学、航空航天、地质勘探、公安、边防、美术设计、游戏、网购等国民经济和社会生活的各个领域。数字图像处理的作用和魅力正日益显现，它几乎吸引了从科学家到平民百姓的普遍关注。可以预见，随着人工智能时代的到来，数字图像处理技术的应用将会更为普及和深入。

1.1 数字图像处理的产生

21世纪是一个智能信息的时代，大数据研究表明，人们日常生活中所接收的信息总量，80%左右来源于图像信息，从这一角度看，"百闻不如一见"正是图像处理重要性的形象表达和经验总结。因此，图像是人类获取信息、表达信息和传递信息的重要手段，是人类感知和认识世界的基础。

图像处理技术的早期应用较为单一，图像处理的目的是改善和提高图像质量，即主要考虑以人为对象，以改善人的视觉效果为目的。在图像处理中，输入的是质量低的图像，输出的是改善质量后的图像。常用的图像处理方法有图像增强、复原、编码、压缩等。现代数字图像处理技术始于20世纪60年代，它受益于微电子技术的快速发展、计算机运算速度的迅速提升和快速算法的出现，从而使得图像处理算法可以在计算机上真正实现。这一时期，美国NASA开始处理飞船从月球发回的照片，标志着数字图像处理技术开始在军事和尖端科技行业得到实际应用。

首次大规模成功应用图像处理技术的是美国喷气推进实验室(JPL)。1964年，美国JPL实验室在图像处理的研究和应用方面进行了开拓性的工作。图像处理工程师利用"徘徊者7号"发回的数千张月球照片，使用计算机及图像处理设备，在充分考虑了太阳位置和月球环境对照片影响的基础上，采用灰度变换、去噪声、几何校正、傅里叶变换以及二维线性滤波等方法对图片进行处理，由计算机第一次绘制了月球表面地图。随后，1965年又对"徘

徊者 8 号”发回的几万张照片进行解卷积和消除运动模糊等较为复杂的处理，使图像质量进一步提高，由计算机成功地绘制出月球表面地图。后来又对探测飞船发回的近十万张照片进行更为复杂的图像处理，获得了月球的地形图、彩色图及全景镶嵌图，为人类登月创举奠定了坚实的基础，也推动了数字图像处理这门学科的诞生。从此，数字图像处理被广泛应用于航空、航天以及遥感遥测等尖端领域并发挥着越来越重要的作用。

在我国，图像处理技术给人们带来巨大的震撼也常发生在航天领域。虽然我国航天技术一直走在世界前列，但一直在低调中奋进，2013 年 6 月 11 日“神舟十号”飞船成功发射，2017 年 4 月 20 日“天舟一号”货运飞船在文昌航天发射中心由长征七号遥二运载火箭成功发射升空。正是数字图像处理技术，使人们近距离感触到了我国航天技术的先进水平，即便人们足不出户，也可以详细地观看到航天员杨利伟、聂海胜、张晓光、王亚平、刘洋在“神舟”系列、“天舟”系列飞船上进行太空探索和科学试验活动。

数字图像处理技术的另一个重要应用是医学领域。1972 年，英国 EMI 的工程师 Housfield 发明了用于头颅诊断的 X 射线计算机断层摄影装置，即通常所说的 CT (computer tomograph)。CT 的基础方法是根据人的头部截面的投影，经计算机处理来重建截面图像，称为图像重建。1975 年，EMI 又成功研制出全身用的 CT 装置，可获得人体各个部位清晰的断层图像。由于图像处理技术这一应用的重大意义，这一无损伤诊断技术的英国研究者 Housfield 和美国研究人员 Commack 获得了 1979 年诺贝尔奖，这说明图像处理技术对人类做出了划时代的贡献。

美国在卫星遥感、军事、气象等领域开始广泛应用数字图像处理技术，快速推动了数字图像处理技术的发展，而这一时期，也正是全球微电子技术走上快速发展的初期，包括 CPU 处理器、海量存储器、图像数字化和图像显示以及计算机硬件设备不断降价，客观上推动了该技术的发展和深入应用。也正是这一时期，美国开始将图像处理技术从军事领域扩展应用到了工业和民用领域，开始利用卫星云图进行天气预报，并利用卫星遥感监测全球气候，在公众安全领域开始使用指纹识别、人脸识别技术，数字图像处理技术为美国社会带来了巨大的经济和社会效益。

借助于信息技术和图像处理技术的领先优势，20 世纪 90 年代，美国提出了“数字地球”的概念。显然，图像是构成“数字地球”的信息基础，不同的遥感平台提供的多源遥感图像为数字地球这一概念提供了基础信息支撑。今天，数字图像处理技术已经应用到科学研究的众多领域，数字图像处理技术对科学研究、国民经济甚至人类社会的发展具有深远的意义。

首先，图像处理技术是人类视觉延伸的重要手段。数字图像处理技术从宏观上扩展了人类视觉可触及的距离。2017 年 4 月 27 日，我国“天舟一号”货运飞船实现了与“天宫二号”的首次推进剂在轨补加试验，借助于图像处理与通信技术，人们坐在家中就能观看到这一尖端科技壮举，看到了我国航天事业的巨大成就。

其次，数字图像处理技术拓宽了人类获取信息的渠道和窗口。对于人眼来说是模糊的或者不可见的图像，通过图像增强技术或其他处理方法，可以使模糊图像变得清晰明亮，使不可见图像变为视觉可见图像。人类视觉系统只能看到电磁波谱中很窄范围的可见光部分，紫外波段、红外波段和微波等波段对人眼都是不可见的，但以现阶段的科技水平，却可以利用红外、微波等波段的信息进行数字成像，通过数字图像处理技术将不可见的波谱信息变为可见图像信息。现代医学也正是利用了数字图像处理技术，将不可见的 X 射线进行成

像。遥感遥测技术也是非常典型的例子，陆地卫星的 TM 多光谱图像就充分利用了波长为 0.78 ～0.94μm 的近红外波段来探测植被信息和生长情况，获取植被宏观信息；利用波长为 10.4～12.5μm 的热红外波段可监测地表大气层的热源污染情况，无论是近红外波段还是热红外波段，都是不可见波谱信息。

数字图像处理技术可以提高人类对信息的识别精度和识别能力。通过数字图像处理技术，可以将人眼无法识别的图像进行识别和分类处理。指纹识别是一个典型的例子，人眼很难区分识别不同个体的指纹，而计算机通过图像识别技术却可以快速准确地检索、匹配和识别数十亿级别的不同个体的指纹。

此外，数字图像处理技术可以帮助人们更加客观、准确、快速地了解世界和认识世界。人类的视觉系统可以帮助人类从外界获得 80％左右的信息，而图像、图形是表示和传递信息的最佳载体。现阶段，绝大多数人都没有登上过月球，但通过“嫦娥三号”月球探测器以及“玉兔号”月球车从月球传回的高清晰月球图像，人们能准确地了解了月球的概貌。

今天，数字图像处理技术已经广泛、深入地应用于国民经济的各个领域。在医学和生物工程领域，X 射线、CT、核磁共振等技术已经广泛用于临床诊断；在军事、航天和国土资源监测领域，卫星遥感图像技术已广泛应用于情报获取以及所有与地球资源相关的矿产勘探、资源监测、水土流失等方面的监测；在公共安全领域，指纹识别、人脸识别、虹膜识别技术已得到广泛的应用，数字安保系统已经在联合国及奥运会等各种重要场所发挥着不可替代的重要作用。

1.2　数字图像处理的基本概念

人出生以后，一睁开眼睛就开始接收各种各样的图像信息，因此有人说，图像与生俱来是人类生活中最直观、最丰富和最生动的信息表示形式。但对于数字图像处理学科而言，图像的定义并非是不言自明的，为研究和讨论的需要，应建立一个科学、统一的概念。

1.2.1　图像及其类型

数字图像处理(digital image processing)又称为计算机图像处理，它是指通过计算机对图像进行去除噪声、增强、变换、复原、分割、特征提取、识别等运算与处理。

图像包括各种各样的形式：可见图像和非可视图像；抽象图像和实际图像；适于和不适于计算机处理的图像。因此，不同种类的图像大量存在于人类生活之中。然而尽管大家天天接触图像，也知道什么是图像，但对图像却没有严格的定义。

在韦氏(Webster)英文词典中，图像(image)一词的英文定义是“an imitation or representation of a person or thing，drawn，painted，photographed…”，即图像是人或事物的一个逼真模仿或描述……著名图像专家 Castleman 博士将图像定义为“在一般意义下，一幅图像是一个物体或对象(objects)的另一种表示”。例如，美国总统布什的照片是他某时刻出现在镜头前的一种描述或表示。因此，图像是其所表示物体或对象信息的一个直接描述和浓缩表示。简而言之，即图像是物体在平面坐标上的直观再现。因此，一幅图像包含了所表示物体的描述信息和特征信息，或者说图像是与之对应的物体或对象的一个真实表示，这个表示可以通过某些技术手段实现。

日常生活中可能会遇到各种各样的图像，如非光学数字图像、高维图像(维数等于或大于 3)、多光谱图像、非均匀采样图像、非均匀量化图像等。图像的分类方法也多种多样，通常情况下，图像可根据其形式或产生方法进行分类。目前被一致接受的分类方法是 Castleman 博士所采用的基于集合论的分类方法，如图 1-1 所示。如果考虑所有物体的集合，图像便形成了其中的一个子集，在图像子集中的每幅图像都和它所表示的物体存在对应关系。根据图像的产生方法，图像可以分为如下三类。

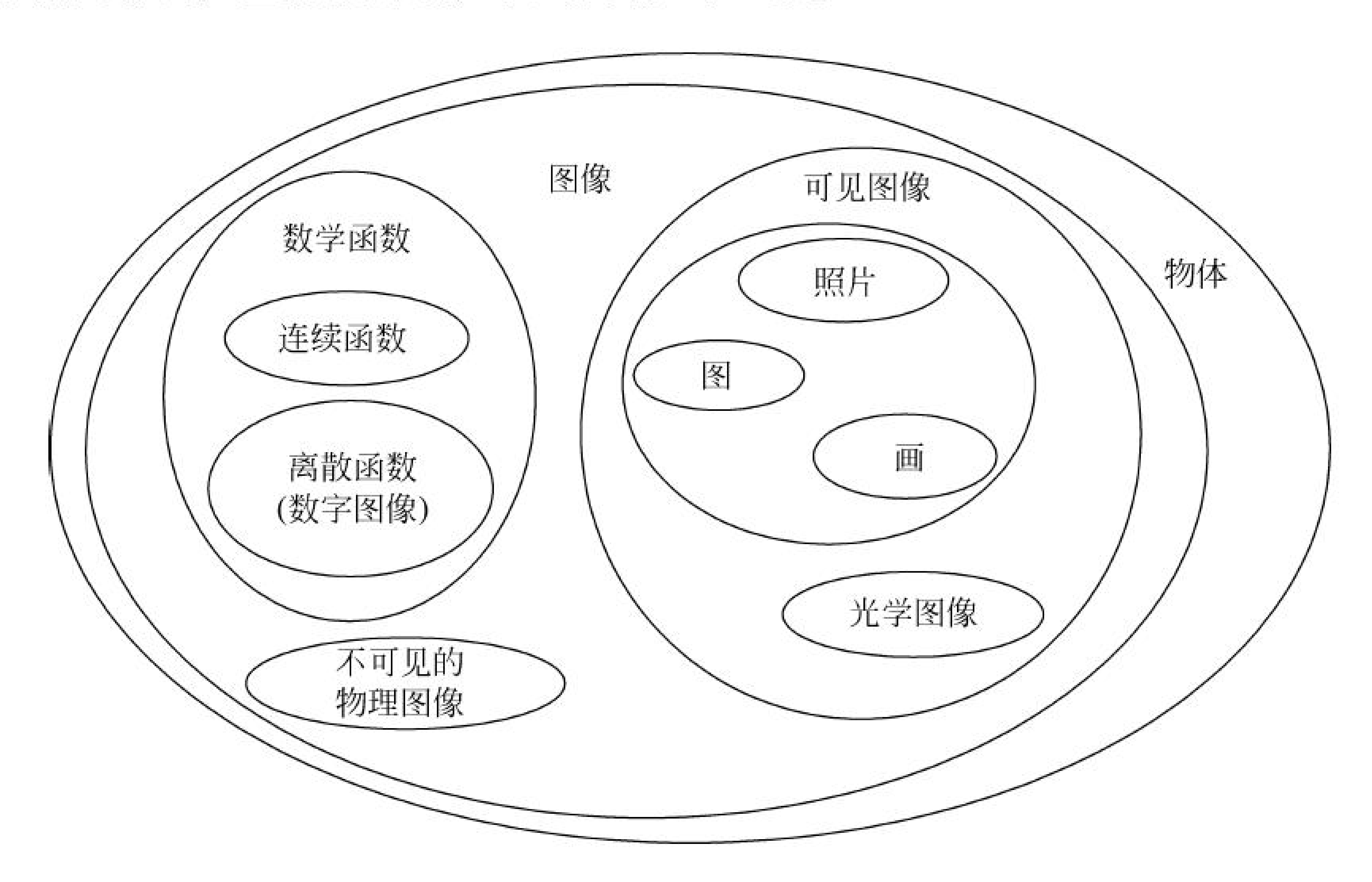


图 1-1　图像的基本类型

1. 可见图像

可见图像(visible images)是指视觉系统可以直接看见的图像，这也是大多数人在日常生活中所见到的和所理解的图像，这一类图像一般通过照像、手工绘制等传统方法获得，通常计算机不能直接处理，但经过数字化处理后可变为数字图像。在该集合中又包含几种不同方法产生的图像子集，其中一个子集为图片(picture)，包括照片(photograph)、图(drawing)和画(painting)；另一个子集为光学图像(optical image)，即用透镜、光栅和全息术产生的图像。

2. 物理图像

物理图像(physical image)所反映的是物体的电磁波辐射能，包括可见光和不可见光图像。例如，光学图像是光强度的空间分布，它们能被肉眼所看到，因此也是可见的图像。不可见物理图像的例子包括温度、压力、高度以及人口密度等的分布图。物理图像一般可以通过某一种光电技术获得，第一类图像中的照片也可以归入此类。物理图像的一个重要子集是多光谱图像，包含物体的近红外、中红外、热红外等波谱信息。绝大部分的物理图像也是数字图像。多光谱图像的每一个点所包含的不只是一个局部特性，如红、绿、蓝三光谱图像，它的技术原理普遍用于彩色照相技术和彩色电视等实际应用当中。黑白图像在每个点只有一个亮度值，而彩色图像则在每个点具有红、绿、蓝三个亮度值，这三个值表示在不同波段上的强度，就是人眼看到的不同颜色。

3. 数学图像

数学图像(maths image)指由连续函数或离散函数生成的抽象图像。其中离散函数所生成的图像就是计算机可以处理的数字图像。

1.2.2　图像与数字图像

图像是其所表示物体的信息的直接描述和概括表示。一般而言,由于图像传感器的性能和成像角度原因,一幅图像所包含的信息应比原物体所包含的信息要少,因此一幅图像并非是该物体的一个完全、精确的表示,但却是一个直观和恰当的表示。

以数学方法描述图像时,图像可以认为是空间各个坐标点上光照强度的集合。也就是说,从物理、光学和数学的层面看,一幅图像可以被看作物体辐射能量的空间分布,这个分布是空间坐标、时间和波长的函数,即

$$I = f(x, y, z, \lambda, t) \tag{1-1}$$

对于静止图像,式(1-1)与时间 t 无关;对于单色图像,式(1-1)与波长 λ 无关;对于二维平面图像,式(1-1)与空间的第三维坐标变量 z 无关。

因此,一幅二维平面图像可以用二维光辐射强度的分布函数(也称为亮度函数)表示,即

$$I = f(x, y) \tag{1-2}$$

如果说图像是与之对应的物体的一个表示,那么数字图像可以定义为一个物体的数字表示,或者说是对一个二维矩阵施加一系列的操作,以得到所期望的结果。其中,数字指采用数字方法对离散像素单元进行的有关处理和计算,像素即离散的单元,处理是指让某个事物经过某一过程的作用,过程即指为实现期望目标而进行的一系列操作。

因此,数字图像处理可以定义为对一个物体的数字表示施加一系列的操作,以得到所期望的结果。其过程表现为由一幅图像变为另一幅经过修改或改进的图像,是一个由图像到图像的过程。而数字图像分析则是指将一幅图像转化为一种非图像的表示。在图片的例子中,处理的过程就是改变图片使其达到某一预定目标,或使其更满意或更具吸引力。通常情况下,若无特别声明,数字图像是指一个被采样和量化后的二维强度函数,采用等间距矩形网格采样,对幅度进行等间隔量化。也可以说,一幅数字图像是一个被均匀采样和均匀量化(即离散处理)的二维数值矩阵。

1.2.3　其他相关概念

计算机图形学(computer graphic)是指利用计算机技术将概念或数学描述所表示的物体(非实物)图像进行处理和显示的过程。其侧重点在于根据给定物体的数学描述模型生成一幅图像。从一定意义上说,如将计算机图像学和图像分析相比较,那么两者处理的对象和输出结果正好相反,计算机图形学是将非图形式的数据描述生成图像。

计算机视觉(computer vision)是指利用计算机、传感器及相关设备模拟生物的视觉功能,主要目的是通过对传感器采集到的图像或视频信息进行处理,理解自然场景的二维或三维信息。在机器人领域中,计算机视觉为机器人提供类似人类视觉的功能。

图像的运算(operation)可分为三类。第一类是全局运算,此类运算是对整幅图像进行相同的处理。第二类是点运算,其输出图像每个像素的灰度值只依赖输入图像对应点的灰度值。点运算有时称为对比度运算或对比度拉伸。第三类是局部运算,它的输出图像上每个

像素的灰度值是根据输入图像中以对应像素为中心的邻域中多个像素的灰度值计算出来的。

噪声(noise)一般是指加性的或乘性的非图像本源信息,即图像的污染信息。

像素(pixel)是指在由一个数字序列表示的图像中的一个最小单位,称为像素。

对比度(contrast)是指一幅图像中灰度反差的大小。

采样密度(sampling density)是指在图像上单位长度所包含的采样点数。采样密度的倒数即像素间距(pixel spacing)。

放大率(magnification)指图像中物体与其所对应的实际景物中物体的大小比例关系。该定义只能用于线性几何关系,即在图像和景物中可以定义相同的测量单位,而且这个比例在全图中是一致的。

数字化(digitizing)是将一幅图像从其原来的形式转换为数字形式的处理过程。"转换"是非破坏性的。从计算机信息处理的角度看,所谓数字图像可以理解为对二维函数 $f(x,y)$ 进行采样和量化(即离散处理)后得到的图像,因此,通常用二维矩阵来表示一幅数字图像。一幅图像进行数字化的过程就是在计算机内生成一个二维矩阵的过程。数字化过程包括扫描、采样和量化三个步骤。

扫描(scanning)是按照一定的先后顺序对一幅图像进行遍历的过程,如按照行优先的顺序进行遍历扫描,像素(即图像元素)是扫描(遍历)过程中最小的寻址单元。对摄影图像的数字化就是对胶片上一个个微小网格的顺序扫描。上述网格又称为矩形扫描网格或光栅(raster)。

采样(sampling)是指遍历过程中,在图像的每个像素位置上测量其灰度值,即亮度值。采样通常是由光电传感器件完成,它将每个像素点的亮度转换成与其成正比的电压值,采样的结果是得到每一像元的灰度值。

量化(quantization)则是将采样得到的灰度值通过模/数转换等器件转换为离散的整数值。由于数字计算机只能处理数字量,因此必须将连续的测量值转换为离散的整数。

1.2.4 数字图像获取方法

数字图像一般可以通过以下几种途径获取:

(1) 图像传感器。图像传感器的作用是将真实场景的空间信息转换为图像信号,以数据文件的形式保存在存储介质上,并传输至计算机,它是图像处理不可或缺的通用设备。例如,专用图像采集卡通常还具有视频编辑功能,可对数字化的视频信号进行后期编辑处理,如剪切画面、添加滤镜、字幕和音效以及加入各种视频特效等。

(2) 数字化扫描仪。扫描仪的作用是将传统的可见光图像经过数字化处理转换为数字图像。例如,将一幅照片通过扫描仪输入到计算机中,扫描仪所进行的过程实质上就是一个数字化过程。

(3) 应用各种光电转换设备直接得到数字图像。例如,卫星上搭载的推帚式扫描仪和光机扫描仪可以直接获取地表甚至地下物体的图像并实时存入存储器中。此外,侧视雷达也可以直接获取数字图像。

(4) 合成图像。直接由二维离散数学函数生成数字图像。

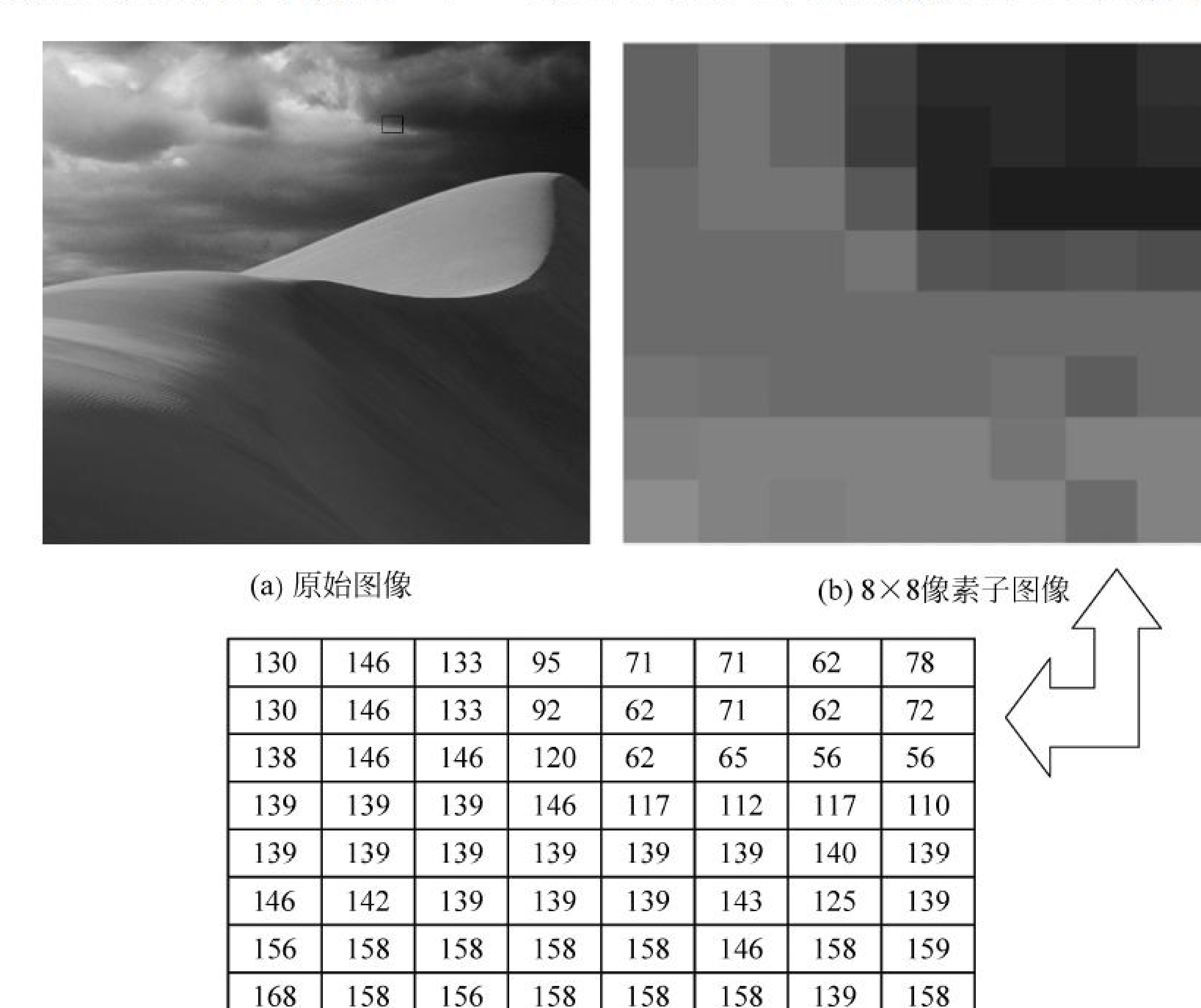
无论采取哪种获取方式,最终得到的数字图像在数学上都是一个二维矩阵。因此,数字图像处理的实质是对二维矩阵进行各种运算和处理。也就是说,将原始图像变为目标图像

的过程，实质上是由一个矩阵变换为另一个矩阵的数学过程。无论是图像的点运算、几何运算、图像的统计特征，还是傅里叶等正交变换，本质上都是基于图像矩阵的数学运算。

因此，数字图像处理是指对一个物体的数字表示，即对一个二维矩阵施加一系列的操作，以得到所期望的结果。

1.2.5　数字图像的矩阵表示

数字图像在计算机中通常采用二维矩阵表示和存储，如图 1-2 所示，形象地描述了由一幅数字图像到该图像所对应的二维矩阵的简易过程和原理。图 1-2(a)所示是一幅大小为 128×128 像素的二维数字图像，为了表述方便，以图 1-2(a)中取出的一个小矩形方块为例，将该小方块放大至像素水平，即图 1-2(b)，可以看出这是原始图像 1-2(a)中的一个 8×8 像素的子图像，放大后的子图像[图 1-2(b)]的每一像素点都具有一个确定的灰度值，将这些灰度值按像素的顺序排列，就是一个二维矩阵。矩阵各元素的值如图 1-2(c)所示。



(a) 原始图像　　(b) 8×8像素子图像

130	146	133	95	71	71	62	78
130	146	133	92	62	71	62	72
138	146	146	120	62	65	56	56
139	139	139	146	117	112	117	110
139	139	139	139	139	139	140	139
146	142	139	139	139	143	125	139
156	158	158	158	158	146	158	159
168	158	156	158	158	158	139	158

(c) 8×8像素子图像的二维矩阵

图 1-2　数字图像的矩阵表示

上述由图 1-2(a)到图 1-2(b)的过程中，原始图像在水平方向和垂直方向被等间隔地分割成大小相同的小方格(grid)，其中的每一个小方格称为像素点，简称为像素或像元(pixel)。像素是构成图像的最小基本单位，图像的每一像素都具有独立的属性，其中最基本的属性包括像元位置和灰度值两个属性。位置由像元所在的行和列的坐标值决定，通常以像素的位置坐标(x,y)表示，像元的灰度值即该像素对应的光学亮度值。

因此，对一幅图像按照二维矩形扫描网格进行扫描的结果是生成一个与原图像相对应的二维矩阵，且矩阵中的每一个元素都为整数，矩阵元素(像素)的位置则由扫描的顺序决

定，每一个像素的灰度值通过采样获取，然后经过量化得到每一像素亮度（灰度）值的整数表示。因此，一幅图像经数字化后所得到的数字图像，实际上就是图 1-2(c)所表示的一个二维整数矩阵，矩阵的大小由图像像素的多少决定。

1.3 数字图像处理系统

计算机技术的快速发展推动了数字图像处理技术的发展，反过来，数字图像处理技术的广泛应用又推动了计算机和微电子技术的发展。图像处理的大数据量，既推动了大容量存储器研发，也推动和培育了海量存储市场的发展。数字图像的高性能显示和输出，促进了高精度彩色显示器和彩色打印机的诞生与发展，并推动了对扫描仪、数码相机、摄像机等高精度数字图像输入设备的发展。

数字图像处理技术的快速发展和广泛应用推动了图像处理系统硬件的研发，不同行业对数字图像处理系统的性能提出了不同的要求。例如，医学影像处理系统对系统的处理速度和精度方面的要求较高，而指纹识别系统则要求系统具有实时响应的性能。虽然不同的图像处理系统在精度、速度、容量等方面的要求可能不尽相同，但数字图像处理系统呈现出专业化的发展趋势。

1.3.1 数字图像处理硬件系统

一个经典的数字图像处理系统如图 1-3 所示，主要包括输入设备、主机系统、输出和存储设备等三部分。

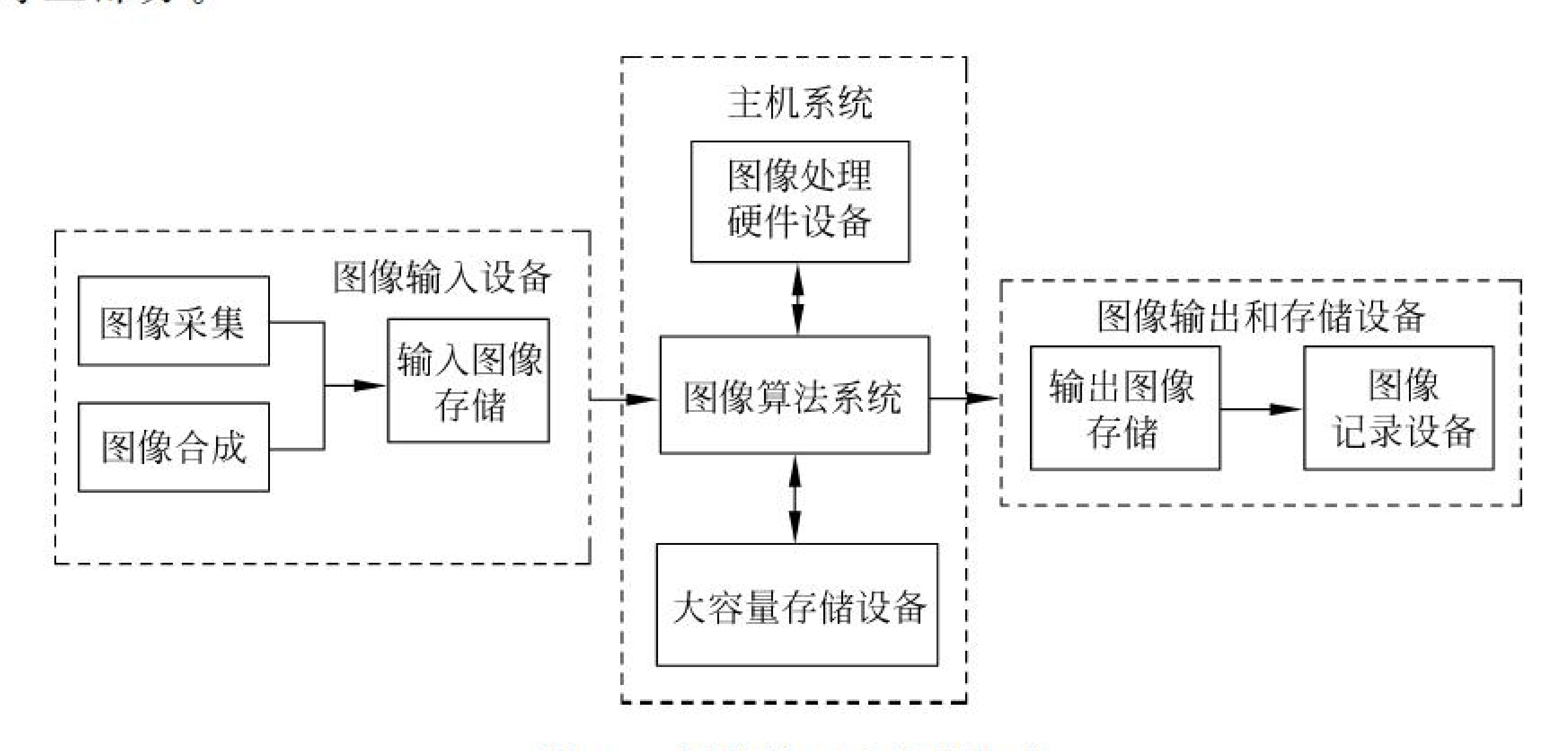


图 1-3 图像处理系统的构成

1. 图像输入设备

图像输入设备包括图像采集与合成设备，以及输入图像存储设备等。随着技术的发展，图像输入设备的性能越来越高，但价格却越来越低。常用的输入设备主要包括图像采集卡、工业摄像机、光电扫描仪、数字相机、遥感遥测等多种设备类型。

(1) 图像采集卡：图像采集卡(image capture card)，又称为图像捕捉卡，是一种可以获取数字视频信息，并将其存储、播放、传输到计算机主机的硬件设备。采集卡是进行图像处理必不可少的硬件设备，很多图像采集卡能在捕捉视频信息的同时获得伴音，使音频部分和视频部分在数字化时同步保存、同步播放。

(2) 光电扫描仪：包括平板式扫描仪和滚筒式扫描仪两种主要类型。

(3) 数字照相机：又称为数码相机，可以进行景物拍摄并以数字形式存储所拍摄的图像。数字照相机的核心部件是电荷耦合器件(CCD)图像传感器，目前主流型号相机的分辨率已超过2000万像素，可以通过USB接口与计算机相连，将图像数据传入计算机并进行进一步处理。

(4) 电子成像管摄像机：电子成像管摄像机的关键部件是摄像器件。摄像器件将输入的二维物体信息(即光学图像)转换为适于处理和传输的电信号。

(5) 固态摄像机：包括CCD阵列、电荷注入器件(CID)阵列和光电二极管阵列三种类型，其中CCD已得到广泛应用，并已逐渐替代电子成像管摄像机。

2. 图像算法系统

数字图像处理的算法系统一般由台式计算机、笔记本电脑、服务器以及其他各种高性能主机系统组成，包括硬件和软件两大部分。常见的图像处理系统主要有计算机和苹果机系统。

(1) 计算机：计算机包括台式计算机、笔记本电脑、服务器以及其他各种高性能主机。个人计算机(PC)通常以Windows或Linux作为操作系统，目前通用的图像处理软件一般都可以在PC上运行。随着微型机速度和容量的提高，PC与小型机工作站的性能差异已经越来越小，包括ERDAS和PCI Works等许多遥感图像处理软件已经可以在PC上运行。随着PC处理速度和存储性能的提高，许多大型的专业化的图像处理系统也逐渐向PC上发展，但20世纪70年代以前，遥感图像处理系统一般只能在工作站上运行。

(2) 苹果机系统：苹果机需使用专用的操作系统软件，在诸如广告、动画制作等领域主要使用苹果机等专业图像处理系统。在图像处理的早期，还有各种形式的图形工作站，如Sun、SGI和HP等公司生产的工作站。工作站一般采用UNIX操作系统，适应于包括遥感图像在内的大型图像处理系统的开发与应用。现在，由于计算机运算速度的快速提升，专业的工作站已很少见。

3. 输出与存储

图像处理系统的输出设备主要有如下三种类型。

(1) CRT显示器：图像处理系统中应用最广泛的输出设备是基于光栅扫描的阴极摄像管所构成的CRT监视器。随着技术的进步，液晶显示器(LCD)、PDP显示器等输出设备也开始得到了广泛的应用。

(2) 打印机：显示器一般用于图像的暂时显示，而打印机可用于将图像永久地记录在纸上。目前常用的图像打印设备主要包括彩色喷墨打印机和激光彩色打印机等。

(3) 其他设备：数字图像还可以通过一些专门的记录设备将图像记录在胶片(底片)上，然后通过传统的洗像技术或打印机输出。大幅面的遥感卫星图像通常采用这种输出形式。

由于数字图像处理系统数据量非常庞大，因此图像处理系统的存储设备一般选用磁带、光盘和大容量硬盘等存储器。

1.3.2　数字图像处理软件系统

数字图像处理系统不仅包含硬件设备，而且需要一定的软件环境支持。以现阶段的情况看，图像处理软件平台一般在Windows下开发，图像软件开发的工具多种多样，但主要包括以下几种。

1. MATLAB 图像工具箱

MATLAB 是由 MathWorks 公司推出的用于数值计算的综合工具。它具有强大的矩阵运算和大数据处理功能，而数字图像处理与矩阵运算具有密切的联系，并且 MATLAB 图像处理工具箱提供了丰富的图像处理函数。熟练使用这些函数可以完成几乎所有的图像处理工作。不断推出的 MATLAB 新版本其图像处理功能越来越强大，应用它进行数字图像处理可以节省大量编写底层算法源代码的时间，避免程序设计中的重复劳动，达到事半功倍的效果。MATLAB 的不足之处是运行程序需要安装有 MATLAB 系统，应用不方便，执行速度较慢。

2. Visual C++ 开发系统

由于 Visual C++ 是一种具有强大综合软件开发能力的系统，而且所开发出来的应用程序运行速度快，并具有一定的可移植性能，因此，Visual C++ 在数字图像处理、分析和识别等方面的应用较为广泛。此外，各国的科学家和研究机构开发了不少专用的图像处理软件环境。

3. OpenCV

OpenCV 的全称是 Open Source Computer Vision Library。OpenCV 于 1999 年由 Intel 公司推出，现由 Willow Garage 提供支持。OpenCV 是一个基于(开源)发行的跨平台的计算机视觉库，可以运行在 Linux、Windows 和 Mac OS 上。Open 属于轻量级且高效的开发工具，它由一系列 C 函数和少量 C++ 类构成，同时提供了 Python、Ruby、MATLAB 等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。目前较新的版本是 2.4.8。目前 OpenCV 拥有包括 500 多个 C 函数的跨平台的中、高层 API，它不依赖其他的外部库(尽管也可以使用某些外部库)。OpenCV 为 Intel Integrated Performance Primitives (IPP)提供了透明接口，这意味着如果有为特定处理器优化的 IPP 库，OpenCV 将在运行时自动加载这些库。OpenCV 提供的视觉处理算法非常丰富，又由 C/C++ 开发，加上其开源的特性，可以无须添加新的外部支持就可以完整地编译、连接生成执行程序，所以很多开发者用它进行算法移植，OpenCV 的代码经过适当改写可以正常运行在 DSP 系统和单片机系统。

此外，国内外的一些研究机构还推出了许多其他数字图像处理软件，如 AVS 系统、SPIDER 系统和 IUE 系统等。AVS 的优点是具有对海量数据进行可视化处理的强大功能，在医学、图像显示、资源探索等方面得到广泛应用。SPIDER 由日本通产省工业技术研究院开发和推出，包含了 FFT、WHT 等许多图像处理领域的基本算法，受到了业界的广泛好评。IUE 是美国、欧洲、日本共同开发的图像处理系统，该系统具有严密的几何学描述，可用于各种类型的图像处理。

1.4 数字图像处理的应用与发展

20 世纪 60 年代以来，数字图像处理技术始终保持着迅速发展的趋势，而且其高速发展势头一直得以延续，其源动力表现在以下几方面：第一，图像处理所需的计算机设备、数字化设备和图像显示设备在不断降价，处理器和大容量存储器末端市场价格逐年下降，而性能却越来越高；第二，与图像处理相关的基础科学研究快速发展；第三，数字图像处理的应用

领域持续扩大，并不断涌现出新的应用。以美国和欧洲的工业化国家为例，尽管总军费开支趋于平稳甚至下降，但在“勇气号”和“机遇号”火星探测器、军事以及航天遥感等领域却更多地使用了数字图像处理技术，而且在商业智能、人工智能、生物医学、大数据等领域对图像技术的应用需求一直持续增长。

1.4.1　数字图像处理的应用

图像是人类获取和交换信息的主要来源，相应地，数字图像处理的应用领域也几乎涉及人类生活、工作的方方面面。随着人类活动范围的不断扩大，图像处理的应用领域也将随之不断扩大。近年来，数字图像处理的应用领域无论深度上还是广度上都发生了深刻的变化。如表 1-1 所示，图像处理技术已经渗透到民用工业、军事、航空航天、医疗保健、环境保护、矿产资源、工程技术、安全保卫和科研等各领域，几乎涉及社会经济生活的各方面，在国民经济中发挥越来越重要的作用。

表 1-1　图像处理的应用领域

应用领域	应用内容
工业	工业探伤、工件识别、机器人控制、质量监测及铁路运输与调度等
军事	导弹制导、夜视瞄准、电子沙盘、军事侦察与训练、军事演习等
航空航天	航天发射、空间站建设与维护、陆地卫星、星际探测与研究等
经济	身份认证、指纹技术、虹膜识别、防伪、网络购物等
通信	传真、视频、多媒体通信、GIS、视频会议等
气象与环保	卫星云图分析、天气预报、水环境监测、大气污染监测与调查等
水利	河流分布、洪水监测、水下泥沙监测与预测、水利及水害调查等
生物医学工程	X 射线照片分析、CT、核磁成像分析、细胞分析、染色体分类等
法律	视频监控、生物特征识别、指纹识别等
物理、化学	结晶分析、谱分析等
地质	地图绘制、矿产监测、油气资源勘探等
海洋	海岸线监测、海洋污染监测、鱼群探查、海洋安全监测等
农林	植被分布调查、农作物估产、病虫害监测、森林火灾监测等

1. 航空航天

数字图像处理技术在航空航天领域的应用非常广泛，除了 JPL 对月球、火星照片的处理之外，还包括很多方面，如 20 世纪 60 年代末以来，美国及一些国际组织发射了数量可观的资源遥感卫星（如 LANDSAT 系列）和天空实验室（如 SKYLAB），由于成像条件受飞行器位置、姿态、环境条件等影响，图像质量不是很高，而必须采用数字图像处理技术。如 LANDSAT 系列陆地卫星，采用多波段扫描器（MSS），在 900km 高空对地球每一区域以 18 天为一周期进行扫描成像，其图像地面分辨率相当于十几米数量级（1983 年 LANDSAT-4 分辨率为 30m）。这些图像在空中先处理（数字化，编码）成数字信号存入存储系统中，在卫星经过地面站上空时，再高速传送至地面中心，然后由处理中心分析判读。这些图像无论在成像、存储、传输过程中，还是在判读分析中，都必须应用数字图像处理技术。在我国，无论是“神舟”系列（1～10 号）、“天宫”系列、“天舟”系列、绕月飞船系列的发射，还是“空间站”的建立，以及各型号的运载火箭和飞船的历次发射、状态监控、故障修复及回收，这其中数字图

像处理技术也都发挥了不可替代的重要作用。

此外，包括航空遥感、卫星遥感、空间探测、军事侦察等诸多领域都离不开图像处理技术。很多国家每天派出一定数量的侦察机对地球上感兴趣的地区进行大量的观察和信息传输，获取军事和经济等方面的有用信息。现在，世界各国在进行资源调查(如森林调查、海洋泥沙和渔业调查、水资源调查等)，灾害勘测(如病虫害检测、水火检测、环境污染检测等)，资源勘察(如石油勘探、矿产量探测、大型工程地理位置勘探分析等)，城市规划(如地质结构、水源及环境分析等)，农业规划(如土壤营养、水分和农作物生长、产量的估算等)都大量采用了图像处理技术。十几年来，我国也陆续开展了上述多方面的一些实际应用，并获得了很好的效果，在航空航天和太空星际研究方面，图像处理技术也发挥了相当大的作用。

2. 工业应用

工业应用是数字图像处理技术的重要应用领域之一。从 20 世纪 60 年代开始，在美国、日本及欧洲的一些工业化国家就已经开始采用数字图像处理技术进行工业生产的质量控制。例如，在现代化的流水生产线上，可以利用图像处理技术对产品和部件进行无损检测；在浮法玻璃生产线上，可以对玻璃质量进行监控和筛选等。

在工业和工程领域中，图像处理技术有着广泛的应用，如在自动装配线上检测零件的质量及对零件进行分类，印制电路板疵病检查，弹性力学照片的应力分析，流体力学图片的阻力和升力分析，邮政信件的自动分拣，在一些有毒、放射性环境内识别工件及物体的形状和排列状态，先进的设计和制造技术中采用工业视觉等。其中值得一提的是，研制具备视觉、听觉和触觉功能的智能机器人将会给工农业生产带来新的激励，目前已在工业生产中的喷漆、焊接、装配中得到有效的利用。

3. 医学与生物工程

由于数字图像处理技术具有直观、无创伤、经济和安全方便等诸多优点，因此图像处理在医学领域的应用非常广泛。从近 30 年来国内外医学领域的发展情况看，图像处理技术已在临床诊断、病理研究等医学领域发挥了重要作用，不仅应用领域日益广泛，而且很有成效。

除了已经提到的获诺贝尔奖的 CT 之外，图像处理技术在医学和生物医学工程领域的应用非常之多。如医用显微图像的处理分析(红/白细胞分类与计数、染色体分析、癌细胞识别等)，目前广泛应用于临床诊断和治疗的各种成像技术，如 X 光肺部图像增强、超声波图像处理、心电图分析、立体定向放射治疗等方面都广泛应用了图像处理技术。

在 CT 推出后，目前类似的设备已有多种，如核磁共振(nuclear magnetic resonance imaging，NMRI)、电阻抗断层图像技术(electrical impedance tomography，EIT)和阻抗成像(impedance imaging)，这些都是利用人体组织的电特性(阻抗、导纳、介电常数)形成人体内部图像的技术。由于不同组织和器官具有不同的电特性，因此这些电特性包含了解剖学信息。更重要的是人体组织的电特性随器官功能的状态而变化，因此 EIT 可望绘出反映与人体病理和生理状态相应功能的图像。目前，EIT 的一些算法正在呼吸系统、消化系统、心血管系统等方面进行临床应用的探索。

4. 通信领域

早在 1865 年，法国在巴黎至里昂就成功试验了传真通信，虽然由于技术及经济原因，当时的图像通信技术发展一直非常缓慢，但开创了人类影像通信的先河。按业务性能划分，图像通信可分为电视广播(点对面通信)、传真、可视电话(点对点通信)、会议电视(单点对多

点)、图文电视、可视图文以及电缆电视等。

当前通信的主要发展方向是声音、文字、图像和数据结合的多媒体通信。具体地讲是将电话、电视和计算机以三网合一的方式在数字通信网上传输。其中以图像通信最为复杂和困难,因图像的数据量十分巨大,如传送彩色电视信号的速率达100Mb/s以上。要将这样高速率的数据实时传送出去,必须采用编码技术来压缩信息的比特量。在一定意义上讲,编码压缩是这些技术成败的关键。除了已应用较广泛的熵编码、DPCM编码、变换编码外,国内外正在大力开发研究新的编码方法,如分行编码、自适应网络编码、小波变换图像压缩编码等。

20世纪70年代,由于微电子技术的突破和大规模集成电路的发展,解决了图像通信中的关键技术,有效推动了图像通信的应用与发展,从此图像通信逐渐成为了人们生活中常用的通信方式。1980年CCITT为三类传真机和公共电话交换网上工作的数字传真建立了国际标准,1984年CCITT提出了ISDN标准,以及当今基于IP的多媒体通信都意味着非语音通信业务在通信中所占据的重要位置。图像通信主要包括如下几种方式:

(1) 传真。传真指将文字、图表、照片等静止图像通过光电扫描的方式变成电信号加以传送的设备。CCITT于1980年颁布了三类数字传真国际标准:一类机不进行压缩,4线/毫米,传送一页A4文件需6分钟;二类机采用了频带压缩技术(残留边带传输),4线/毫米,传送一页A4文件需3分钟;三类机在传送前采用了去冗余技术,在电话线上传送一页A4文件需1分钟。四类机是在三类机的基础上发展起来的传真设备,采用了去冗余、纠错码技术,连接Modem以后可以在公用电话网上使用。经过多年发展,传真技术不断进步,目前的分辨率可达16点/毫米以上,传送一幅A4文件仅需数秒。

(2) 电视广播。1925年英国实现了单色电视广播,1936年英国BBC正式开始推出电视广播。目前彩色电视主要有三种制式,即美国、日本等国所采用的NTSC制式,中国和西欧、非洲等地区所采用的PAL制式,以及法国、俄罗斯等国所采用的SECAM制式。

(3) 图文电视和可视图文。图文电视(teletext)和可视图文(videotext)是提供可视图形文字信息的通信方式。图文电视是单向传送信息,是在电视信号消隐期发送图文信息,用户可用电视机和专用终端收看该信息。可视图文是基于双向工作方式的,用户可用电话向信息中心提出服务内容或从数据库中选择信息。

(4) 有线电视。有线电视即CATV,它是采用电缆或光缆传送电视节目的电视技术。1949年,美国安装了全球第一个有线电视系统,1977年采用光缆实现了有线电视。随后,有线电视系统由于其信号稳定性、信号质量和频道数量等方面的特点在全球得到迅速普及。

(5) 可视电话和会议电视。自从1964年美国国际博览会首次展出了带宽为1MHz的picture-phone MOD-I可视电话系统以来,可视电话和会议电视系统由于其具有方便和高效等特点而得到广泛应用。目前,可视电话和会议电视均采用数字压缩技术,并采用H.261、H.263、H.230、H.323等相关图像编码和会议电视国际标准。

5. 遥感

1962年国际上才开始正式使用遥感(remote sensing)一词。可以说数字信号处理技术对遥感遥测技术的发展与推动起了非常重要的作用,遥感技术发展的重要事件无不与数字图像处理技术密切相关。1909年意大利人乘飞机拍摄世界上第一张航空照片、1957年苏联和1958年美国发射的第一颗人造地球卫星都采用了大量的数字图像处理技术。此后,美国

相继发射多颗陆地资源探测Ⅱ卫星。

在航空遥感和卫星遥感技术中，某些军事大国每天派出很多侦察飞机对地球上的敏感地区进行大量的空中拍摄，对拍摄的照片进行处理分析，以前需要雇用几千人，而现在改用配备有高级计算机的图像处理系统来判读分析，既节省了人力，又加快了速度，还可以从照片中提取人工所不能发现的大量有用情报。1982 年，美国基于 LANDSAT-N 卫星成功组网定位精度达到±10m 级的全球定位系统(global positioning system，GPS)。

2017 年 11 月 5 日，中国第三代导航卫星顺利升空，标志着我国正式建成北斗全球卫星导航系统(BeiDou navigation satellite system，BDS)。这是我国自行研制的全球卫星导航系统，是继美国 GPS、俄罗斯格洛纳斯卫星导航系统(GLONASS)之后第三个成熟的卫星导航系统。北斗全球卫星导航系统由空间段、地面段和用户段三部分组成，可在全球范围内全天候、全天时为各类用户提供高精度、高可靠定位、导航、授时服务，并具有短报文通信能力，已经具备区域导航、定位和授时能力，定位精度小于 10m，测速精度为 0.2m/s，授时精度为 10ns。未来，基于图像处理技术的遥感遥测技术效率及分辨率越来越高，应用越来越普及和深入，在土地测绘、资源勘探、军事侦察、气象监测与预报、林业资源勘测、环境污染监测、农作物估产等方面将得到更广泛的应用。

6. 安全与交通

目前，在安全领域已开始采用图像处理技术与模式识别等方法实现重要场所、特定现场和敏感地点的监控，可以通过采用足迹、指纹识别、人脸识别、虹膜识别等技术实现案件侦破，通过数字图像技术和通信技术实现对交通的管理与调度等。

7. 军事和公安

在军事方面图像处理和识别主要用于：导弹的精确末制导；各种侦察照片的判读；具有图像传输、存储和显示的军事自动化指挥系统，飞机、坦克和军舰模拟训练系统等；公安业务图片的判读分析，指纹识别，人脸鉴别，不完整图片的复原，以及交通监控、事故分析等。目前已投入运行的高速公路不停车自动收费系统中的车辆和车牌的自动识别都是图像处理技术成功应用的例子。

8. 金融与支付

21 世纪是信息化时代，基于互联网、电子商务、在线购物的应用日益普及，金融与电子支付已广泛使用指纹密码、在线身份认证、产品防伪、数字证书、人脸识别甚至虹膜识别等技术，这些无不与图像处理技术密切相关。

9. 视频和多媒体

目前，视频和多媒体系统已广泛应用于电影与电视制作中，许多电视制作人员都能熟练使用图像处理、变形、合成等技术制作各种特技动作。多媒体系统离不开静止图像和动态图像的采集、压缩、处理、存储和传输。

图像处理技术的应用领域已非常广泛，它在社会经济发展、国计民生、国防与国家安全以及日常生活中发挥着越来越重要的作用。

1.4.2 数字图像处理的发展

始于 20 世纪 60 年代现代数字图像处理技术，随着微电子技术发展、计算机运算和处理速度的提升、各种快速算法的出现，已经从最初的航天探测等少数尖端领域向现代文明的各

个方向渗透，图像处理技术将向高速、高分辨率、多媒体、智能及标准化方向发展，主要表现在以下几个方面。

1. 高速、高分辨率和高精度

数字图像处理技术在进一步加强理论研究，逐步形成图像处理学科体系的同时，应进一步提高硬件速度，这不仅要提高计算机的速度，以及 A/D 和 D/A 的速度和精度；而且应该提高分辨率，主要包括采集分辨率和显示分辨率，着重解决图像处理速度等核心问题，将图像、图形技术相结合，朝着三维成像或多维成像的方向发展。例如，在航天遥感、气象预报的云图处理方面，巨大的数据量和处理速度仍然是主要矛盾之一。

2. 新理论与新算法

近年来，数字图像处理领域在理论上也有了更新的发展，新的理论和新的算法不断涌现，如 Wavelet 算法、Morphology 算法、Fractal 算法、分形几何(Fractal)、数学形态学神经网络算法、遗传算法、卷积神经网络等。其中 Fractal 算法广泛用于图像处理、图形处理、纹理分析，同时还可以用于数学、物理、生物、神经和音乐等方面，这些理论及建立在其上的算法，将会成为今后图像处理理论与技术的研究热点，并将进一步推动图像处理技术的发展。此外，图像处理技术将加强新算法的研究与加强软件研究相结合，开发新的处理方法，特别要注意移植和借鉴其他学科的技术和研究成果，创造新的处理方法。

3. 与通信技术结合

数字图像处理技术在理论、软件、硬件技术不断发展的同时，其另一个新的发展方向是图像与通信技术的紧密结合。从 1865 年的传真通信开始，到目前正开始进入应用的可视电话，图像处理与通信技术的结合已经历了一百多年的历程。随着微电子技术、计算机技术，特别是 VLSI 制造技术、通信数字化技术的迅速发展，图像与通信技术日益融合。会议电视、电视电话、图文电视、可视图文、传真等图像通信方式已应用到各行各业。20 世纪 90 年代初，以 H. 261(活动图像编码国际标准)以及随后一系列图像编码、图像通信的国际标准先后获得通过为标志，解决了可视技术在通信中的应用这一长期困扰人们的问题，极大地推动了会议电视、电视电话等图像通信方式的国际化和产业化，图像通信开始进入一个高速发展的新阶段。

4. 图像处理的硬件技术

在图像处理技术方面，一个新的趋势是更加重视图像处理的专门硬件芯片的研究，把图像处理的众多功能固化在芯片上，使之更加便于应用。围绕 HDTV(高清晰度电视)的研制，开展实时图像处理的理论及技术研究，向着高速、高分辨率、立体化、多媒体化、智能化和标准化方向发展。20 世纪 80 年代后期开始，图像处理的硬件技术也得到了迅速发展，这时不仅能处理二维图像，而且开始进行三维图像的处理。目前，一些图像处理硬件采用流水线结构，可以将 JPEG 集成到一个芯片上。进入 21 世纪以来，图像处理技术已逐步渗透到人类生活和社会发展的各个方面。例如近年来蓬勃发展的医学图像处理、多媒体信息处理技术、图像融合技术、虚拟现实技术等，图像在其中均占据了主要地位，文本、图形、动画、视频都要借助于图像处理技术才能充分发挥它们的作用。

5. 图像处理领域的标准化

图像的信息量大、数据量大，因而图像信息的建库、检索和交流是一个重要的问题。就现有的情况看，软件与硬件种类繁多，交流和使用极为不便，成为资源共享的严重障碍。应

建立图像信息库，统一存放格式，建立标准，统一检索方法。此外，还应加强边缘学科的研究工作，促进图像处理技术的发展。如人的视觉特性、心理学特性等方面的研究，如果有所突破，将对图像处理技术的发展起到极大的促进作用。

6. 新的应用领域

从JPL实验室开拓性地引入数字图像处理技术到我国北斗全球卫星导航系统以及空间站的建立，都标志着数字图像处理技术为人类的航空航天及军事等尖端领域做出了持续的贡献，与此同时，美国、日本和欧洲一些国家的科技人员又将图像处理技术从空间技术推广到了生物学、医学、光学、陆地卫星、多波段遥感图像分析、人工智能、粒子物理、地质勘探、工业检测及印刷等多种应用领域。特别是进入21世纪以来，关于图像处理方面的研究论著，在质量和数量上都在迅速攀升，技术上的突破不断促进数字图像处理技术向深度和广度发展，图像水印、图像检索等各种新的应用方向不断出现。

数字图像处理理论和技术经过了约50年的快速发展，已不再局限于航空、航天、通信和医学等少数尖端领域，已迅速发展成一门独立的具有强大生命力的学科，并渗透到了科学研究的各个领域、工业生产的众多行业、人类生活的各个方面。随着各种应用需求的不断增加，数字图像处理无论在理论上还是实践上都存在着巨大的潜力，可以预料，数字图像处理技术必将更加迅速地向广度和深度发展。今天，随着科技事业的进步以及人类需求的多样化，多学科的交叉融合已是现代科学发展的突出特色和必然途径，而图像处理学科是一门既与尖端科学紧密相连，又与国计民生密切相关的应用科学，必将产生更多的应用领域和更新的成果。

习题

1. 什么是图像？图像与数字图像有何区别？
2. 简要分析为什么在计算机系统中以二维矩阵表示数字图像。
3. 什么是物理图像？物理图像包括哪几种类型？
4. 什么是数字图像？在计算机中，数字图像如何表示和存储？
5. 在升学和工作的常规体检中，经常要进行X射线照片和胸透等方面的检查，分析这些医学成像设备采用了哪些数字成像技术，这些技术对医学检查和诊断具有哪些作用。
6. 某个图形工作环境使用32×32像素的图像作为图标(icon)，每个像素具有16种颜色，总共可产生多少种不同的图标？
7. 若某一大小为200×200像素的图像，每个像素的灰度有40级，假设每1亿个可能图像中仅有1个是可识别的，观看每一幅图像需用时1s，那么看完全部图像需多长时间？
8. 图像处理系统通常包含哪些部分？各部分有何作用？
9. 应用MATLAB开发数字图像系统具有哪些优点？
10. 数字图像处理技术的发展方向主要包括哪些方面？

第2章 数字图像处理基础

CHAPTER 2

日常生活中，人们睁开眼睛就可以看到各种各样的图像，这些图像包罗万象，既包括各种海报、平面杂志图片、报刊图片等模拟图像，也包括计算机网络、电子文档中的数字图像。对于数字图像，还可按图像类型和格式类型进行进一步划分。众所周知，计算机只能处理数字信息，因此数字图像处理首先需将模拟图像转换为以数字形式表示和储存的数字图像，同时，对于数字图像，根据科学研究、工业生产和日常其他需要，有时还需要对数字图像进行质量评价或其他统计评价。

本章主要对数字图像的类型、图像数字化器、图像数字化中的采样和量化原理、像素间的基本关系、常用图像文件格式、图像的显示与打印、图像质量评价方法、灰度直方图以及数字图像的统计特征等方面进行了分析和讨论。

2.1 数字图像的类型

在计算机中，为方便对数字图像进行处理，静态数字图像根据其特性可以分为矢量(vector)图和位图(bitmap)，位图也称为栅格图像。

矢量图是采用一系列绘图指令来表示一幅图，这种方法的本质是以数学和几何学中的公式描述一幅图像，图像中每一个形状都是一个完整的公式，称为一个对象。对象是一个封闭的整体，所以定义图像上对象的变化以及与其他对象的关系对计算机来说是简单的，所有这些变化都不会影响到图像中的其他对象。数学形式的矢量图文件具有两个显著的优点：数据量很小；图像质量与分辨率无关。无论将矢量图像放大或缩小多少次，图像总是以显示设备允许的最大清晰度显示，在计算机计算与显示一幅图像时，也往往能看到画图的过程。但矢量图也具有明显的缺点，即不易制作色调丰富或色彩变化太多的图像，而且绘出来的图像逼真度差，也不易在不同的软件间交换文件。

位图是计算机表示数字图像最常用的一种类型。根据颜色和灰度级数量，可以将位图分为二值图像、灰度图像、索引图像和RGB彩色图像四种基本类型。目前，大多数图像处理软件都支持这四种类型的图像。

1. 二值图像

二值图像的灰度值只有0、1，其中灰度值0代表黑色，1代表白色。因此，二值图像所对应的二维矩阵元素也由0、1构成。由于每一像素(矩阵中每一元素)的取值只有0、1两种，

所以计算机中二值图像的数据类型采用一个二进制位表示。采用扫描仪扫描图像时，若设置为 LineArt 格式，则扫描仪以一位颜色模式来看待图像。若采样点颜色为黑，则扫描仪将相应的像素位元置为 0，否则置为 1。二值图像通常用于文字、线条图的扫描识别(OCR)和掩膜图像的存储。

二值图像又称为线画稿，即只有黑、白两种颜色，这种形式通常也称为“黑白艺术”“位图艺术”“一位元艺术”。

2. 灰度图像

灰度图像一般指具有 256 级灰度值的数字图像，灰度图像只有灰度值而没有彩色。因此，每个像素都是介于黑色和白色之间的 256 种灰度中的一种。256 级灰度值图像的数据类型为 8 位无符号整数，灰度值 0 表示纯黑色，255 表示纯白色，0～255 之间的数字由小到大表示从纯黑到纯白之间的过渡色。二值图像可以看成是灰度图像的一个特例。

在一些软件中，灰度图像也可以采用双精度数据类型(double)表示，对应的像素值域为[0,1)，其中 0 代表纯黑色，1 代表纯白色，0～1 之间的小数表示不同的灰度等级。

3. 索引图像

索引图像的文件结构与灰度图像和 RGB 图像文件不同，它既包括存放图像数据的二维矩阵，还包括一个颜色索引矩阵(称为 MAP)，因此称为索引图像，又称为映射图像。MAP 矩阵也可以由二维数组表示，MAP 的大小由存放图像的矩阵元素的值域(即灰度值范围)决定。例如，若矩阵元素值域为 0～255，则 MAP 矩阵的大小为 256×3，矩阵的三列分别为 R(红)、G(绿)、B(蓝)值，以 MAP=[R G B]表示。图像矩阵的每一个灰度值对应于 MAP 中的一行，如某一像素的灰度值为 64，则表示该像素与 MAP 矩阵的第 64 行建立了映射关系，该像素在屏幕上的显示颜色由 MAP 矩阵第 64 行的[R G B]叠加而成。也就是说，计算机打开图像文件时，其索引矩阵也同时读入，图像每一像素的颜色以灰度值作为索引，通过检索颜色索引矩阵 MAP 得到实际颜色。计算机中，索引图像的数据类型一般为 8 位无符号整型，即索引矩阵 MAP 的大小为 256×3。因此，一般索引图像只能同时显示 256 种颜色。

与灰度图像一样，索引图像的数据类型也可采用双精度浮点型(double)。索引图像一般用于存放色彩要求比较简单的图像，如 Windows 中色彩构成比较简单的壁纸多采用索引图像存放，如果图像的色彩比较复杂，就要用到 RGB 彩色图像。

4. RGB 彩色图像

RGB 彩色图像又称为真彩色图像，RGB 彩色图像与索引图像都是计算机可以处理的彩色图像。RGB 彩色图像也同样以 R、G、B 三原色的叠加来表示每个像素的颜色。与索引图像不同的是，RGB 彩色图像每一个像素的颜色值(由 RGB 三原色表示)直接存放在图像矩阵中，不需要进行索引。由于数字图像以二维矩阵表示，而每一像素的颜色需由 R、G、B 三个分量来表示，因此 RGB 彩色图像矩阵需要采用三维矩阵表示，即 $M\times N\times 3$ 矩阵，M、N 分别表示图像的行、列数，三个 M×N 的二维矩阵分别表示各个像素的 R、G、B 颜色分量。RGB 彩色图像的数据类型一般为 8 位无符号整型，通常用于表示和存放真彩色图像，当然也可以存放灰度图像。

虽然索引图像和 RGB 彩色图像都可以存放彩色图像，但两者之间数据结构不同，因此存在明显差别。由于索引图像所表示的颜色数量是由索引矩阵 MAP 的大小决定的，而 MAP 的大小又由像素灰度值的值域决定，所以在 8 位无符号整型数据的情况下，索引图像

只能表示 256 种颜色；而 RGB 彩色图像则将每一像素的 R、G、B 三个颜色分量直接存放在三维图像矩阵中，因此理论上其所表示的颜色可多达 2^{24}（$2^8\times2^8\times2^8$）种颜色，远远多于索引图像的 256（2^8）种颜色，而且像素的颜色直接存放在图像矩阵中，在读取数据时无须索引，所以 RGB 彩色图像的显示速度很快。但索引图像也有自己的优点。首先，其所占用的存储空间远远小于 RGB 彩色图像，因此在对图像颜色要求不高的情况下，一般可采用索引图像存放彩色图像。其次，由于索引图像的颜色值存放在索引矩阵 MAP 中，在修改图像的颜色时直接修改索引矩阵即可，而不需要修改图像矩阵，因此在比较选择不同的图像处理方案时，索引图像显得非常方便。

综上所述，可以看出在图像的四种基本类型中，随着图像所表示的颜色类型的增加，图像所需的存储空间逐渐增加。二值图像仅能表示黑、白两种颜色，但所需的存储空间最少；灰度图像可以表示由黑到白渐变的 256 个灰度级，每个像素需要一个字节存储空间；索引图像可以表示 256 种颜色，与灰度图像一样，每个像素需要一个字节存储，而为了表示 256 种颜色，还需要一个 256×3 颜色索引矩阵；RGB 彩色图像可以表示 2^{24} 种颜色，相应的每个像素需要 3 字节的存储空间，是灰度和索引图像的 3 倍。因此，在用高级语言（如 C++）编程存取不同类型图像文件时要注意文件指针的最小寻址单元。表 2-1 给出了四种基本图像类型表示的颜色类型和存储空间的统计（假设图像大小为 $M\times N$）。

表 2-1 四种图像类型的比较

图像类型	二值图像	灰度图像	索引图像	RGB彩色图像
颜色数量	2	256 灰度	256 彩色	2^{24} 彩色
数据类型	1bit	8bit	8bit	24bit
矩阵大小	$M\times N$	$M\times N$	$M\times N$	$M\times N\times3$

对于数字图像处理技术而言，一般都包括以上这四种基本类型图像文件。此外，在多媒体技术中还会遇到高维图像，即图像矩阵的维数大于 3，如多媒体动画图像，由于一个动画图像可包含多帧 RGB 彩色图像，因此动画图像本质上是四维矩阵 $M\times N\times3\times P$，即每一帧图像本身是一个三维矩阵（$M\times N\times3$），而动画彩色图像共有 P 帧 RGB 彩色图像。

2.2 图像数字化器

图像数字化是指将模拟图像信息转换为数字信息的过程，数字相机、扫描仪等设备可以将现实世界中的模拟、连续色调的可视信息转换为计算机可识别的数字信息，这就是数字化过程。图像数字化器必须能够将一幅图像分解为像素并确定每个像素的位置，测量每个像素的灰度级，将所测量的连续亮度信息量化形成数字信息，并将这些数字信息保存在存储设备上。

数字图像处理的早期，图像数字化设备价格非常昂贵，结构复杂，只有少数研究机构才可以使用，随着微电子技术的发展与应用的不断普及，图像数字化设备价格逐年下降，各种各样的设备已经开始应用于图像数字化的处理过程。

2.2.1 图像数字化器的组件

一般而言，一个能将模拟图像信息转换为数字信息的图像数字化器必须具有如下 6 个

基本部分或功能组件。

(1) 人工光源：常见的人工光源主要包括白炽灯、激光器件、荧光物质和发光二极管(LED)等。白炽灯、激光器件是最常用的人工光源。在扫出系统中，物体或图片的数字化使用白炽灯作为光源非常方便，在扫入系统中，一个LED的细丝经一个透镜成像，并形成一个小光点。

(2) 采样孔：使图像数字化器能不受图像其他部分的影响而在整幅图像中扫描特定的独立像素单元。

(3) 扫描器件：其作用是使采样孔以预先确定的方式在图像上移动，按照顺序依次扫描图像的每一个像素。扫描系统包括机械扫描装置、电子束聚焦和集成电路等。如图2-1所示是一个典型的机械扫描装置，一幅照片(图像)部分或全部卷在滚筒上，滚筒通过旋转牵引图像经过固定的采样孔，形成一个方向上的扫描。采样孔安装在引导螺杆上，引导螺杆旋转使采样孔在图像上移动。滚筒和引导螺杆结合形成对二维图像的扫描。如果引导螺杆连续而不是间歇转动，则扫描是螺旋式进行的，但已非常接近直线式扫描。

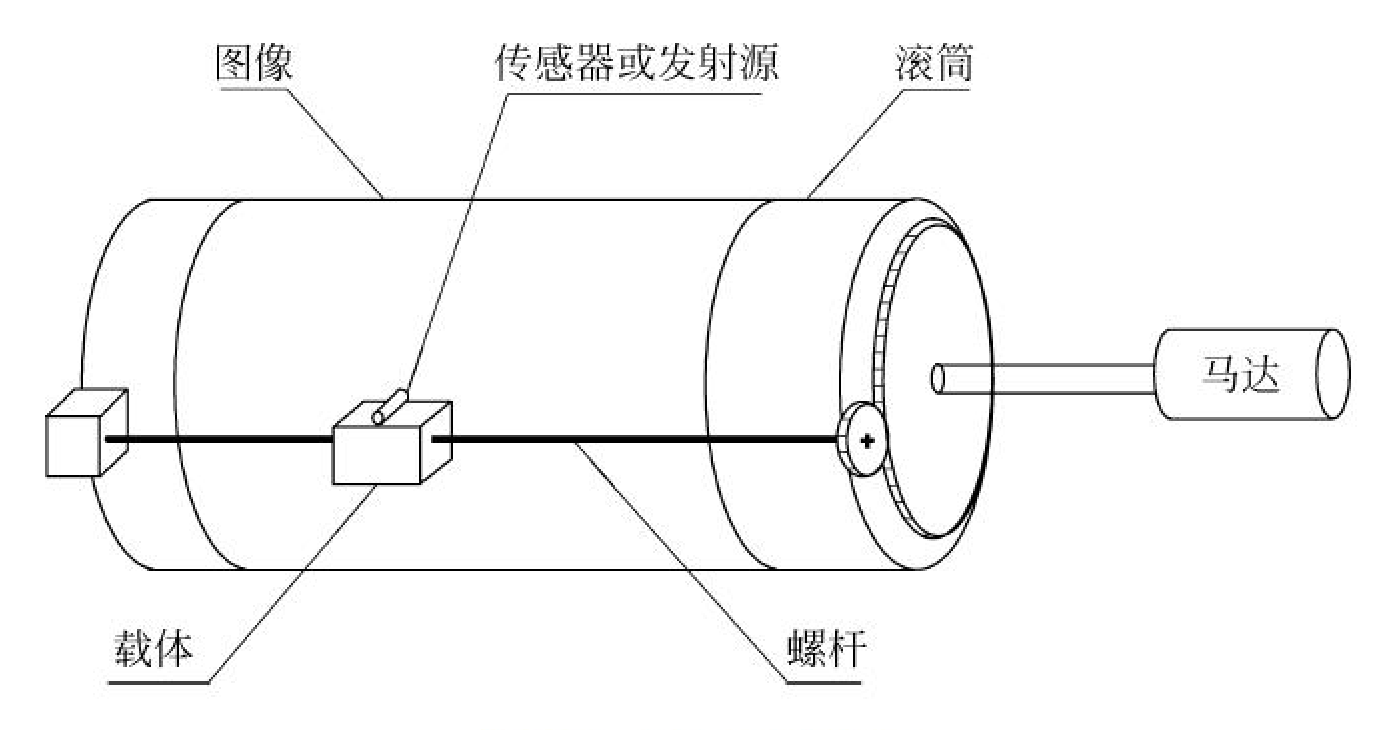


图2-1 机械扫描装置

(4) 光传感器：通过采样孔量测每个像素的亮度信息，将各像素的光强信号转换为电压或电流信号。光传感器的特点是可以产生正比于光强的电信号，光传感器包括光电倍增管、光电二极管、光电晶体管、光敏电阻和固体传感器。

(5) 量化器：将传感器输出的连续信号转换为离散的数字信号(整数值)，数字化部件由模—数转换器电路组成。

(6) 输出介质：将量化器产生的灰度值以适当格式存储起来，以备数字图像处理后续使用。输出介质可以是固态存储器、磁盘或其他合适的存储设备。

2.2.2 图像数字化器的性能

虽然不同的图像数字化器功能不尽相同，但一般而言可以从以下几个方面选择和检查图像数字化器的主要性能和参数。

(1) 像素大小：采样尺寸和相邻像素的间距是两个重要的性能指标，如果图像数字化器安装了放大倍数可调的光学系统，那么输入图像平面上的采样尺寸和间距也是可调的。

(2) 图像尺寸：图像数字化器可处理的最大图像尺寸也是一个重要参数，如胶片扫描仪，最大的输入尺寸是35mm或11×14inch(1inch＝2.54cm)的X光片。对于输出而言，图像尺寸由每行和每列的最大像素数决定。

(3) 物理参数：所谓物理参数，即图像数字化仪器能测量和量化的物理参数。以胶片扫描仪为例，可以测量和量化胶片的透射率或胶片的光密度，这两个参数均为胶片明暗度的函数，但对不同的应用，透射率和光密度的重要程度可能不一样。

(4) 线性度：线性度是指对光强进行数字化时，灰度正比于图像亮度的实际精确程度，图像数字化器的线性度是一个重要的性能指标，非线性的数字化器会影响后续处理的有效性。

(5) 灰度级：能将图像量化为多少级灰度是图像数字化器的重要性能指标之一。图像数字化经历了早期的黑白两级灰度、灰度图像(8 位 256 灰度级)、彩色图像及现在的 RGB 彩色图像。目前已经出现能达到更高分辨率的数字化器。

(6) 噪声：最后一个重要特性是数字化器的噪声水平。如果用数字化器处理一幅均匀的灰度图像，则虽然输入亮度是一个常量，但是系统中的固有噪声却会使输出图像的灰度发生变化。数字化器引入的噪声是图像质量下降的根源之一，应当使噪声小于图像内的反差(即对比度)。

2.2.3　图像数字化器的类型

图像数字化器主要包括数字相机和胶片扫描仪。

数字相机(包括数字摄像机)是一类种类繁多而又重要的数字化器，它可以数字化镜头前的任意物体，如具有计算机接口的电视摄像机。数字相机不仅能数字化实际的物体，还能数字化相片的底片。

胶片扫描仪也是一种非常重要的图像数字化器。尽管胶片扫描仪目前应用范围有限，但依然是一种很重要的图像数字化器。这类仪器专用于扫描胶片上的摄影图像，首先物体由胶片相机摄影，然后由胶片扫描仪将其图像数字化。胶片扫描仪曾经在图像处理过程中占有突出的地位，但目前人们更倾向于直接使用数字相机。

图像数字化一般有扫入和扫出两种方法。在扫出方式中，光线连续照射整个物体或胶片图像，采样孔每次仅允许光传感器探测到一个像素。扫入方式又称为扫进方式，在扫入方式中，物体每次只有一小点被照亮，所有传过来的光线都被传感器收集，物体由光束扫描，即在这种情况下，是物体被照明光线扫描，而传感器则“看到”整个场景。

2.3　图像的采样和量化

目前的图像获取方式中，很多情况下，传感器输出的是连续变化的电压信号，由此所获得的图像是模拟图像信息。为了产生一幅数字图像，需要进行采样与量化，即对模拟图像在空间(x,y)方向上以及亮度函数 $f(x,y)$都进行离散化处理，通常把这一过程称为采样与量化。如何合理地进行图像的采样与量化是关系数字图像与原图像是否接近的两个重要因素，也关系最后形成的数字图像信息量的大小。

2.3.1　采样

图像在空间(x,y)上的离散化称为采样，具体地说，就是以空间上部分点的灰度值代表一幅图像，而这些点称为采样点。由于图像是一种二维分布的信息，因此，为了对它进行采

样操作，需在垂直方向和水平方向分别进行采样。具体做法是，先沿垂直方向按一定间隔从上到下顺序地沿水平方向直线扫描，取出各水平线上灰度值的一维扫描信息。而后再对一维扫描线信号按一定间隔采样得到离散信号，即先沿垂直方向采样，再沿水平方向采样，这样就完成了二维图像的数字化采样操作。对于运动图像，即时间域也是连续的图像，则需先在时间轴上采样，再沿垂直方向采样，最后沿水平方向采样。

如果对图像按照等间距进行采样，每行（即横向）取 M 个点，每列（即纵向）取 N 个点，则图像被排列为 $M\times N$ 的矩阵，矩阵中的每一个点对应于数字图像中的一个元素，称为像素或像素点。因此，该图像也称为具有 $M\times N$ 像素的图像。采样点的多少和采样间隔的选取是一个非常重要的问题，它直接影响到采样后图像的质量，即数字图像与原图像的失真程度。以一维信号为例，若一维信号 $f(t)$ 的最大频率为 ω，根据采样定理

$$f(t)=\sum_{i=-\infty}^{\infty}f(iT)s(t-iT) \tag{2-1}$$

式中

$$s(t)=\frac{\sin(2\pi\omega t)}{2\omega t} \tag{2-2}$$

以 $T\leqslant 1/(2\omega)$ 为间隔进行采样，能够根据采样结果 $f(iT)(i=\cdots,-1,0,1,\cdots)$ 完全恢复 $f(t)$。对于同样采样点数，采样过程还可以分为均匀采样和非均匀采样。

在对图像进行采样的过程中，若在 (x,y) 方向上均进行等间距的采样，则称为均匀采样，反之，则称为非均匀采样。

非均匀采样是指在图像的不同区域，根据图像的具体情况进行不等间隔的采用。非均匀采样的间隔选取要依据原图像中包含的细微浓淡变化情况决定。一般地，图像中细节越多，采样间隔越小。因此，一般在图像细节少的区域采用比较稀疏的采样，在细节变化大的区域采用较密的采样，这样，所获得的图像有用信息量并没有减少，但总数据量却有效地降低了。需要指出的是，在分配采样点时，应在灰度变化的边界上记录下非均匀采样的边界。例如，若图像由均匀的背景上放置的一束花或其他物体组成，则背景可采用稀疏采样，而花的部分则应采用较密的采样，并标出非均匀采样的边界。

2.3.2 量化

模拟图像经过采样以后，在时间和空间上被离散化为像素，但采样所得到的像素值依然是连续量。量化过程就是以离散的灰度值信息代替连续的模拟量灰度信息的过程，该过程是一对多的映射过程。量化可以分为线性量化和非线性量化两种方式。

1. 线性量化

模拟图像的亮度值 f 是连续变化的数值，若 $f(x,y)$ 的亮度值 L 的范围为 $[L_{\min},L_{\max}]$，则称区间 $[L_{\min},L_{\max}]$ 为灰度级范围（或色度范围）。如图 2-2 所示，若将灰度值区间 $[L_{\min},L_{\max}]$ 分成 K 个等间距的子区间，则称为线性量化或等间隔量化。

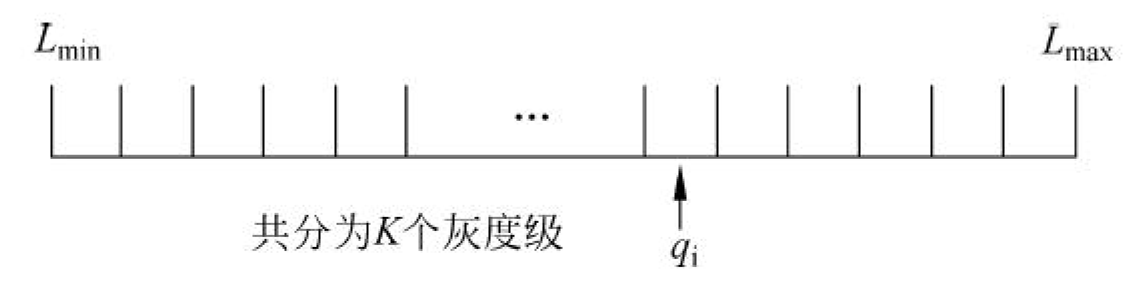
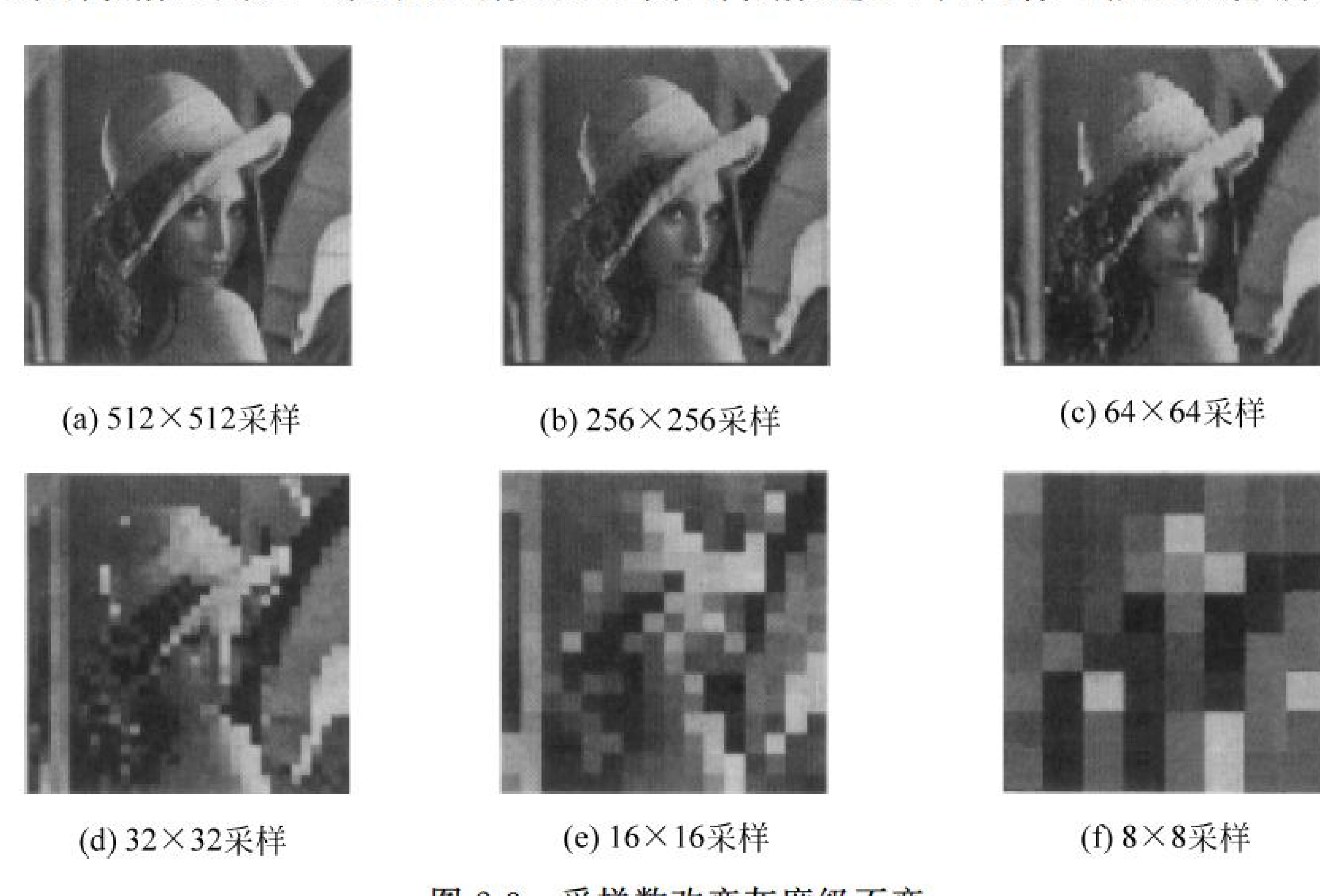


图 2-2 线性量化示意图

量化过程中每个子区间对应一个亮度值 q_i，这样在灰度值范围$[L_{\min}, L_{\max}]$内就有 K 个亮度值对应，称为灰度级 K。为了计算机处理的方便，灰度级 K 一般以 2 的整数次幂表示，即 $K=2^m$，$m=1,2,\cdots,8$ 等，若 $m=8$，则 $K=256$，表示共有 256 个灰度级。

当 $f(x,y)$的连续亮度值被 K 个离散的亮度值取代，对于第 i 个点，取值为 q_i，如何表示每一个采样点的亮度值也是最终影响数字图像质量的关键因素之一。因此，灰度级 K 的选择决定了数字图像的质量，也决定了数字图像所占计算机存储空间的大小。例如，一幅单色调灰度级为 64 的 128×128 的图像，需要 128×128×6=98 304 位表示。$2^6=64$，即位深度为 6，位深度越深，能表示的灰度级就越多。一般情况下计算机表示单色调图像普遍采用 256 个灰度级，因此位深度为 8。如果是一幅 128×128 的彩色图像，每个 RGB 颜色分量都用 256 个灰度级表示，需要 128×128×8×3 位表示，即每一个像素实际上需用 24 位表示。

图 2-3 所示是在 256 级灰度不变的情况下，不同采样点数的数字图像质量对比。图 2-3(a)为 512×512 采样，图 2-3(b)为 256×256 采样，图 2-3(c)为 64×64 采样，图 2-3(d)为 32×32 采样，图 2-3(e)为 16×16 采样，图 2-3(f)为 8×8 采样。可以看出，同样灰度级下，图像质量会因采样点数的减少，效果逐渐变差，当采样点数过少时，图像可能无法分辨。



(a) 512×512采样　(b) 256×256采样　(c) 64×64采样

(d) 32×32采样　(e) 16×16采样　(f) 8×8采样

图 2-3　采样数改变灰度级不变

同样，若灰度级数量减小，对图像质量也具有明显的影响。如图 2-4 所示，(a)为 256 灰度级，(b)为 128 灰度级，(c)为 64 灰度级，(d)为 16 灰度级。通过该图可以看出，对于数字图像，若采样数不变，当灰度级数量减少到一定程度，对图像质量也将产生一定的不利影响。



(a) 256灰度级



(b) 128灰度级



(c) 64灰度级



(d) 16灰度级

图 2-4　采样数不变灰度级改变

2. 非线性量化

与线性量化的等间隔划分灰度区间相反，若将表示数字图像的灰度级范围分为不等间隔的子区间，则称为非线性量化或非均匀量化。与均匀采样和非均匀采样的概念类似，对于灰度级出现频度高的范围可以选择较窄的量化区间，对于一些灰度级出现频度较低的范围可以选择较宽的量化区间。

以图 2-5 为例，同样量化为 16 个子区域，(a)为等间隔的线性量化，即图像整个灰度区间以相同间隔量化；(b)表示在中间灰度级出现频度高的范围内，量化区间变窄，而两端的灰度级因出现频度较低，量化区间则变宽，从而可以控制中间灰度级部分的图像细节内容的信息量损失程度，因此可以实现以较少的灰度级量化图像，也能尽可能得到较高的图像质量。

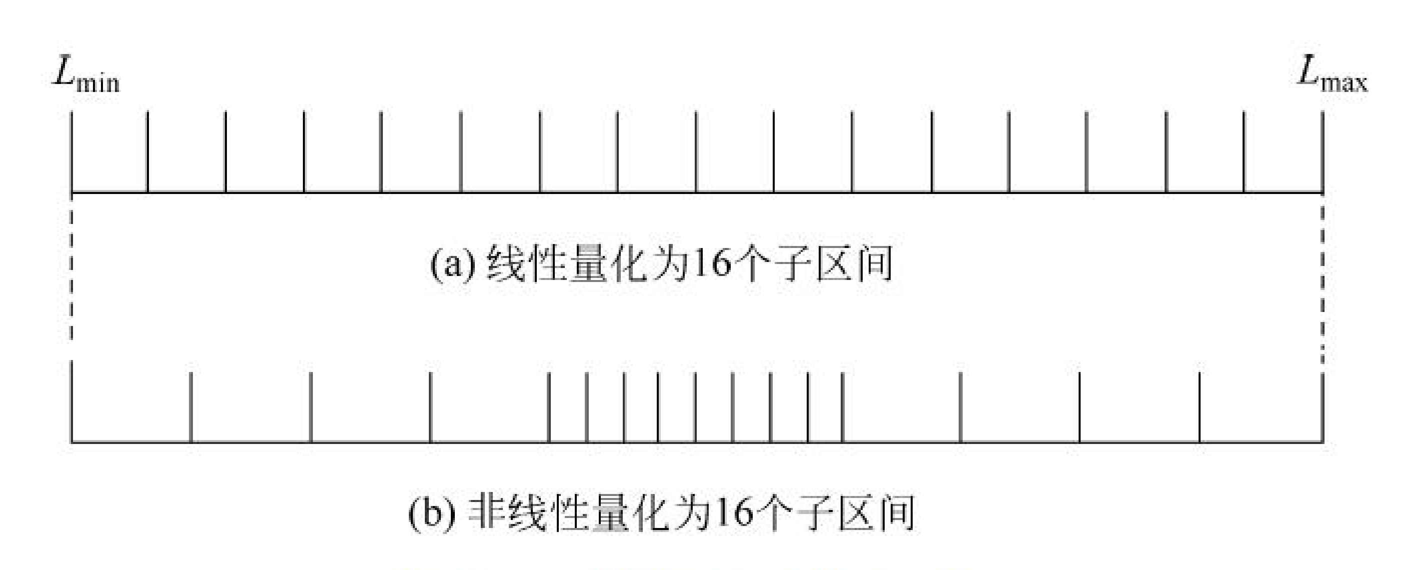


图 2-5 非线性与线性量化对照

2.3.3 采样与量化参数的选择

虽然采样和量化参数都对数字图像的质量具有影响，但实际上两个参数之间的匹配关系也非常重要。并不是一味地提高采样点数和灰度级数量就可以获得高清晰度且高质量的数字图像，在确定采样和量化参数时，还应根据原始图像的性质与质量进行科学、合理的选择。例如，在某些情况下，对细节比较少的图像，当采样点数一定时，图像的质量有可能会随着灰度级的减少而得到一定程度的改善，原因是减少灰度级会增加图像的对比度。

2.4 像素基本关系

进行图像处理和分析时，许多运算只和当前像素的灰度值有关，因此对这些操作只需考虑当前像素的灰度值，如常见的对比度拉伸、直方图均衡化、直方图规定化等运算。但也有一些其他运算和处理需要考虑当前处理像素与其周围相邻像素的关系，如边缘提取、图像分割等运算。本节将介绍在图像处理中经常遇到的像素间基本关系，为后续的学习打下基础。

2.4.1 像素的邻接

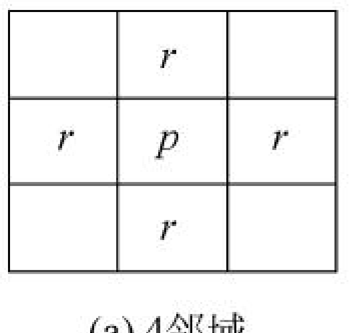
图像处理学科中，像素的邻接表示像素间的空间接近关系。以二维数字图像为例，图像中的每一个像素类似于离散网格中的一个点，即网格中的一个小方格。对于图像中的每一个像素，在空间位置上与它邻接的像素构成它的邻域。设任意像素 p 的坐标为(x,y)，则该像素的 4 邻域定义为其上、下、左、右的 4 个像素如图 2-6(a)所示，这 4 个像素在图中用 r 表示，其坐标分别为$(x,y-1)$、$(x,y+1)$、$(x-1,y)$、$(x+1,y)$。

该像素上、下、左、右的 4 个像素通常记为 $N_4(p)$，在一些参考文献中，该四个位置按方位也可称为东(east)、西(west)、南(south)、北(nonh)。

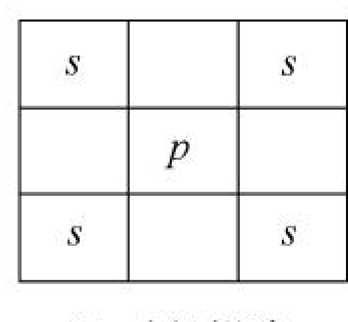
与四邻域类似，当前像素上、下、左、右 4 个方向，加上左上、左下、右上和右下 4 个沿对角线方向的相邻像素，称为当前像素 p 的 8 邻域，如图 2-6(c)所示，记为 $N_8(p)$。

其中，左上、左下、右上和右下 4 个相邻像素的坐标分别为$(x-1,y-1)$、$(x-1,y+1)$、$(x+1,y+1)$、$(x+1,y-1)$。

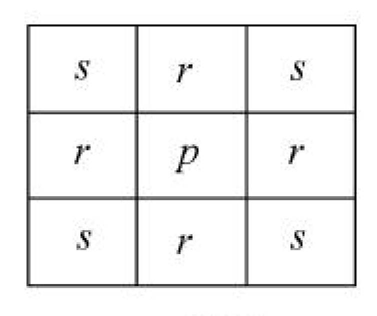
上述 4 个像素(图中以 s 表示)定义为 p 点的对角邻域，记为 $N_D(p)$，如图 2-6(b)所示。在图像分析与处理中，经常使用 4 邻域和 8 邻域的概念，对角邻域的单独应用并不多。



(a) 4邻域



(b) 对角邻域



(c) 8邻域

图 2-6　像素的邻域

2.4.2　邻接性、连通性、区域和边界

1. 邻接性

邻接性是指满足某一个灰度相似性定义的两个像素 p 和 q 是否具有上节所描述的相邻关系中的一种。例如，若定义灰度集合 $C=\{a\leqslant c\leqslant b\}$)，$p$ 和 q 同属于 C，当 p 处在 q 的 4 邻域中时，则称它们为 4 邻接；类似地，如果 p 处在 q 的 8 邻域或对角邻域中，则称它们具有 8 邻接关系，或对角邻接关系。

在介绍邻接性之前，先介绍关于 m 邻接的定义。

m 邻接是指 p 在 q 的 4 邻域中，或 p 在 q 的对角邻域中且 p 和 q 的 4 邻域的交集为空集，即交集中不存在属于集合 C 的像素。

在不同的灰度集合定义下，图像中两个像素的邻接关系可能不一样。如图 2-7 所示，当集合 $C=\{2\}$时，坐标点(1,1)和(2,2)是 m 邻接关系。因为它们同属于集合 C，且它们 4 邻域的交集点(1,2)和点(2,1)不属于集合 C。若所选择的集合 C 为$\{1,2\}$，这时由于坐标点(1,2)的值为 1，属于集合 C，因此，(1,1)和(2,2)就不是 m 邻接关系，但它们均与点(1,2)形成了 m 邻接关系。

	0	1	2	3	4
0	1	2	0	2	2
1	2	2	0	0	1
2	0	1	2	0	1
3	2	1	1	2	2
4	0	0	1	1	2

图 2-7　像素邻接关系图

2. 连通性

当坐标位置为 $p(x1,y1)$的像素与坐标位置为 $q(x2,y2)$的两个像素按某种邻接关系存在一条连通的路径时，称它们是连通的。由像素点 p 到 q 所经历的路径的像素点序列称为由 p 到 q 的路径，从 p 点出发沿路径到 q 点所需走的步数称为路径的长度。

需要指出的是，在分析图像区域的连通性时，一般需要根据邻接关系来确定所考虑的 4 连通、8 连通，还是 m 连通。因为有些像素在 4 邻接条件下是不连通的，但在 8 邻接的条件下则是连通的，如图 2-7 所示，当集合 C 为{2}时，对于坐标点(1,1)和(2,2)，如考虑 8 连通性，则它们是连通的，但对于 4 连通，它们是不连通的。

还是以图2-7为例，当C定义为{2}时，当计算路径长度时，由$p(2,2)$到$q(4,4)$不存在4连通的路径，但存在长度分别为2和3的两条8连通路径，分别是{(2,2)、(3,3)、(4,4)}和{(2,2) (3,3)，(4,3)，(4,4)}，而且存在一条路径长度为3的m连通路径{(2,2)，(3,3)，(4,3)，(4,4)}，而且该路径是唯一的。由此可见，m邻接的特点是，在计算路径长度或描述区域边界时具有唯一性，不存在二义性。

3. 区域和边界

在一幅图像中。由连通的像素点所组成的点的集合称为一个区域。对于区域中的某一像素，如果存在某一个邻域不属于这一区域，则称它是该区域的边界点。一个区域的所有边界点组成该区域的边界，由于这些边界点在区域内，该边界称为区域的内边界。类似地，若边界点并不在当前考虑区域内，但这些边界点有一个邻域属于当前区域，则称为该区域的外边界，所有满足外边界定义条件的像素点组成了区域的外边界。

与连通性相似，区域也分为4连通区域和8连通区域，如图2-7所示，如果集合C为{2}时，根据4连通性，则图2-7中具有5个值为2的区域，坐标点(1,1)和(2,2)分别属于不同的连通区域；如若根据8连通属性，则坐标点(1,1)和(2,2)属于同一个连通区域，图中仅有3个值为2区域。

与此相关的另一个概念是边缘(edge)和边界(border)。边缘是指图像中灰度值存在差异的地方，通常是指相邻像素之间的灰度值差异大于某一个阈值(相关的内容后续章节中将做进一步的介绍)，图像内区域或物体之间的边缘并不一定组成一个闭合轮廓；而边界通常对应于某一个物体的轮廓(contour)，因此边界是闭合的。

2.4.3 距离度量

像素之间的关系与像素在空间上的接近程度相关，像素在空间的接近程度可以用距离进行衡量。众所周知，根据数学知识，距离有多种定义，在图像处理知识体系框架内，距离也包括多种定义。给定三个像素的坐标分别为$p(x,y)$、$q(s,t)$、$r(u,v)$，若满足以下三个基本条件，则度量函数D能称为距离。

(1) 非负性：$D(p,q)\geqslant 0$，当且仅当$p=q$时等号成立。

(2) 对称性：$D(p,q)=D(q,p)$。

(3) 三角不等式：$D(p,q)\leqslant D(p,r)+D(r,q)$。

上述三个条件中，条件(1)保证了距离的非负性；条件(2)表明两个像素点之间的距离与像素点的起终点没有关系；条件(3)表明两个像素点之间的距离，其直线距离最短。在数字图像处理中，距离的定义也必须满足以上三个条件，图像处理常用的距离定义包括以下几种。

(1) 欧氏距离：

$$D_e(p,q)=\sqrt{(x-s)^2+(y-t)^2} \tag{2-3}$$

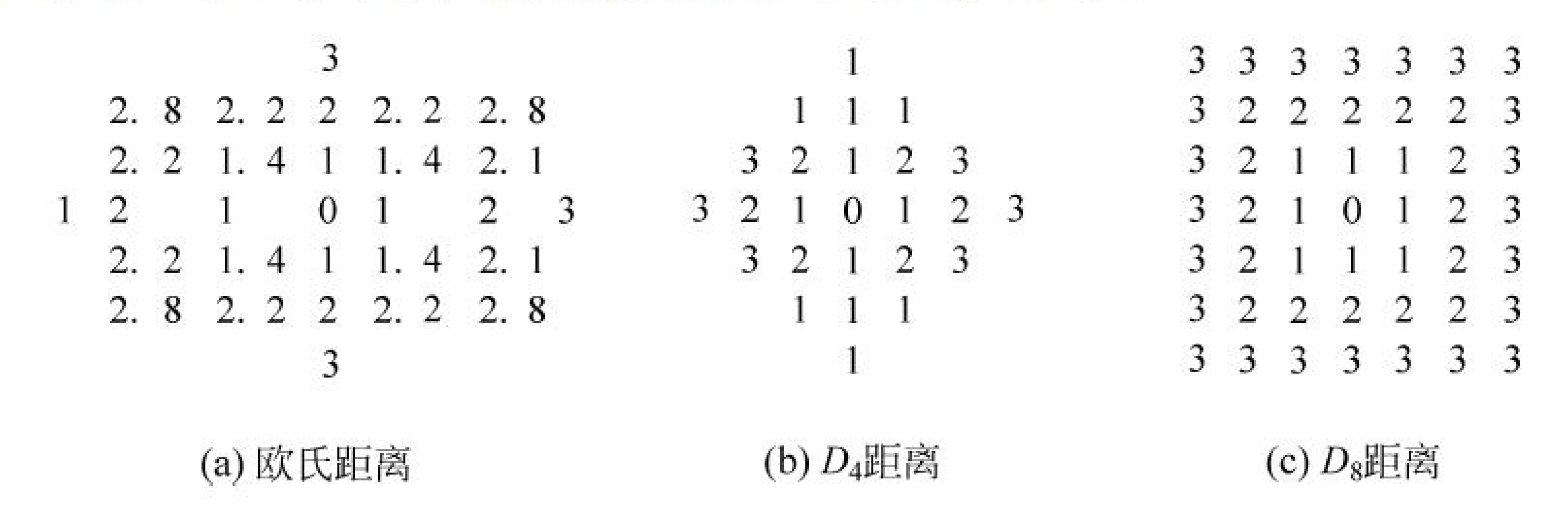
(2) 城市距离：

$$D_4(p,q)=|x-s|+|y-t| \tag{2-4}$$

(3) 棋盘距离：

$$D_8(p,q)=\max(|x-s|,|y-t|) \tag{2-5}$$

图 2-8 给出了中心像素点与周围像素的 D_4 与 D_8 距离图。



(a) 欧氏距离　(b) D_4距离　(c) D_8距离

图 2-8 等距离轮廓示意图

2.5 图像文件格式

图像文件格式是记录和存储影像信息的格式。对数字图像进行存储、传输、处理，必须采用一定的图像格式，也就是把图像的像素按照一定的方式进行组织和存储，将图像数据存储为文件就成为图像文件。图像文件格式决定了图像文件的数据结构信息，以及文件如何与其他文件交换数据等。常见的图像格式文件包括 BMP、JPEG、PNG、TIFF、GIF、RAW 等。由于数字设备获取的图像文件一般都很大，而储存容量却有限，因此图像通常都会经过压缩再储存。

2.5.1 BMP 文件

BMP 是英文 Bitmap（位图）的简写，即位图文件，是一种与硬件设备无关（device-independent Bitmap）的图像文件格式。BMP 文件是 Windows 系统中的标准图像文件格式，被很多 Windows 应用程序支持。随着 Windows 系统的流行及应用程序的不断丰富，BMP 位图格式得到了越来越广泛的应用。BMP 文件采用位映射存储格式，其特点是包含的图像信息较丰富，除位图深度可选之外，不进行任何其他压缩，但由此也导致了它与生俱来的缺点是占用磁盘空间过大。BMP 图像文件的深度可以选择 1bit、4bit、8bit、24bit。采用该格式存储图像时，BMP 图像的扫描按照从左到右、从下到上的顺序进行。

BMP 图像文件格式一般由文件头信息块、图像描述信息块、颜色表（真彩图像无颜色表）和图像数据区四个部分组成，并以 BMP 为扩展名。由于 BMP 文件格式是 Windows 环境中交换与图有关的数据的一种标准，因此在 Windows 环境中运行的图形图像软件都支持 BMP 图像格式。

1. BMP 文件的组成

BMP 文件由位图文件头、位图信息头、颜色信息（调色板数据）和位图数据共四个部分组成，如表 2-2 所示。

表 2-2 BMP 文件结构

BMP 文件组成	数据结构表示
位图文件头	BITMAPFILEHEADER
位图信息头	BITMAPINFOHEADER
颜色信息	RGBQUAD
位图数据	BYTE

2. BMP 文件头

BMP 文件头数据含有 BMP 文件的类型、文件大小和位图起始位置(相对于位图文件头的偏移量)等信息。FILEHEADER 结构占 14 字节,其定义如下:

```
typedef struct tagBITMAPFILEHEADER
{
    WORD bfType;           //位图文件的类型,必须为 BM(第 1、2 字节)
    DWORD bfSize;          //位图文件的大小,以字节为单位(第 3~6 字节)
    WORD bfReserved1;      //位图文件保留字,必须为 0(第 7、8 字节)
    WORD bfReserved2;      //位图文件保留字,必须为 0(第 9、10 字节)
    DWORD bfOffBits;       //位图数据的起始位置,以相对于位图(第 11~14 字节)
                             文件头的偏移量表示,以字节为单位
} BITMAPFILEHEADER;
```

3. 位图信息头

BMP 位图信息头数据用于说明位图的尺寸等信息,主要包括位图大小、像素深度、位图是否压缩等信息。位图信息头共占 40 字节,其定义如下:

```
typedef struct tagBITMAPINFOHEADER
{
    DWORD biSize;            //本结构所占用的字节数(第 15~18 字节)
    LONG biWidth;            //位图的宽度,以像素为单位(第 19~22 字节)
    LONG biHeight;           //位图的高度,以像素为单位(第 23~26 字节)
    WORD biPlanes;           //目标设备的级别,必须为 1(第 27、28 字节)
    WORD biBitCount;         //每个像素所需的位数(第 29、30 字节);设置为 1 表示二值图像,设
                               置为 4 表示 16 色;设置为 8 表示 256 色;设置为 24 表示真彩色
    DWORD biCompression;     //位图压缩类型(第 31~34 字节);设置为 0 表示不压缩,设置为 1
                               表示 BI_RLE8 压缩类型;设置为 2 表示 BI_RLE4 压缩类型
    DWORD biSizeImage;       //位图的大小(其中包含为补齐行数是 4 的倍数而添加的空字节),
                               以字节为单位(第 35~38 字节)
    LONG biXPelsPerMeter;    //位图水平分辨率,每米像素数(第 39~42 字节)
    LONG biYPelsPerMeter;    //位图垂直分辨率,每米像素数(第 43~46 字节)
    DWORD biClrUsed;         //位图实际使用的颜色表中的颜色数(第 47~50 字节)
    DWORD biClrImportant;    //位图显示过程中重要的颜色数(第 51~54 字节)
} BITMAPINFOHEADER;
```

对于 BMP 格式的图像文件,无论文件尺寸的大或小,都不会对数据进行任何压缩处理。如果位图文件采用压缩格式,则 16 色位图采用 RLE-4 压缩算法,256 色位图采用 RLE-8 算法。

4. 颜色表

颜色信息包含本图像用到的颜色表,显示图像时需要用该颜色表来生成调色板。如果图像为 RGB 真彩色图像,图像的每一像素用 24bit 表示,这时图像文件中没有颜色表信息块。颜色表用于说明位图中的颜色,它有若干个表项,每一个表均是一个 RGBQUAD 类型的结构,定义一种颜色。RGBQUAD 结构的定义如下:

```
typedef struct tagRGBQUAD
{
    BYTE rgbBlue;            //蓝色的亮度(范围: 0~255)
```

```
    BYTE rgbGreen;              //绿色的亮度(范围: 0～255)
    BYTE rgbRed;                //红色的亮度(范围: 0～255)
    BYTE rgbReserved;           //保留,必须为 0
} RGBQUAD;
```

颜色表 RGBQUAD 结构数据的个数由 biBitCount 来确定。当 biBitCount=1,4,8 时,分别有 2,16,256 个颜色表项;当 biBitCount=24 时,为真彩色图像,每个字节分别对应 R、G、B 值,此时图像无颜色表项。在 BMP 文件的颜色信息表 RGBQUARD 结构中增加了一个保留字段 rgbReserved,它不代表任何颜色信息,但必须取固定值 0。

根据上述 RGBQUARD 结构对颜色信息的定义,红、绿、蓝排列顺序与一般真彩色图像文件的颜色排列顺序正好相反。即一个位图中某一像素点的颜色为“FF、00、00、00”,则表示该点为蓝色,而不是红色。

位图信息头和颜色表组成位图信息 BITMAPINFO 数据结构,BITMAPINFO 结构定义如下:

```
typedef struct tagBITMAPINFO
{
    BITMAPINFOHEADER bmiHeader;                //位图信息头
    RGBQUAD bmiColors;                         //颜色表
} BITMAPINFO;
```

5. 位图数据

BMP 文件中的位图数据表示图像对应的像素值。位图数据记录了位图的每一个像素值,图像像素值的记录顺序在扫描行内是从左到右,在扫描行之间是从下到上。即 BMP 文件首先存放的是图像的最后一行像素,最后存储图像的第一行像素值,对于同一行之间的像素值,则按照先左边后右边的顺序存储。

位图的一个像素值所占的字节数:

当 biBitCount=1 时,8 个像素占 1 字节;

当 biBitCount=4 时,2 个像素占 1 字节;

当 biBitCount=8 时,1 个像素占 1 字节;

当 biBitCount=24 时,1 个像素占 3 字节,按顺序分别为 B、G、R 值。

Windows 规定一个扫描行所占的字节数必须是 4 的倍数(即以 long 为单位),不足则以 0 填充,位图每一扫描行的字节数:

DataSizePerline=(bi.biWidth×bi.biBitCount+31)/8

位图数据的大小:DataSizePerline×biHeight

若 BMP 文件起始部分数据如下:

424D 46900000 0000 0000 4600 0000 2800 0000 8000 0000 9000 0000 0100 * 1000 0300 0000 0090 0000 A00F 0000 A00F0000 0000 00000000 0000 * 00F8 E007 1F00 0000 * 02F1 84F1 04F1 84F1 84F1 06F2 84F1 06F2 04F2 86F2 06F2 86F2 86F2…

其中,1～14 字节为图像文件头,数据所表示的含义分别如下。

(1) 1、2:(这里的数字代表的是字节,下同)图像文件头。0x4d42='BM',即 BMP 格式。

(2) 3～6：整个文件大小。4690 0000，即 00009046h＝36934。

(3) 7、8：保留，必须设置为 0。

(4) 9、10：保留，必须设置为 0。

(5) 11～14：从文件开始到位图数据之间的偏移量(14＋40＋4 * 2 ^biBitCount))(在有颜色板的情况下)。4600 0000，为 00000046h＝70，上面的文件头就是 35 字＝70 字节。

15～54 为位图信息头数据，数据含义如下。

(6) 15～18：位图信息头长度。

(7) 19～22：位图宽度，以像素为单位。8000 0000，为 00000080h＝128。

(8) 23～26：位图高度，以像素为单位。9000 0000，为 00000090h＝144。

(9) 27、28：位图的位面数，该值总是 1。0100，为 0001h＝1。

(10) 29、30：每一像素的位数。1(单色)，4(16 色)，8(256 色)，16(64K 色，高彩色)，24(16M 色，真彩色)，32(4096M 色，增强型真彩色)。1000 为 0010h＝16。

(11) 31～34：压缩信息说明，即 0(不压缩)，1(RLE 8，8 位 RLE 压缩)，2(RLE 4，4 位 RLE 压缩，3(Bitfields，位域存放)。

(12) 35～38：用字节数表示的位图数据的大小，该数必须是 4 的倍数，数值上等于一行所占的字节数×位图高度。0090 0000 为 00009000h＝80×90×2h＝36864。假设位图是 24 位，宽为 41，高为 30，则该数值＝(biWidth * biBitCount＋31)/32 * 4 * biHeight，即＝(41 * 24＋31)/32 * 4 * 30＝3720。

(13) 39～42：水平分辨率，像素/米。A00F 0000 为 0000 0FA0h＝4000。

(14) 43～46：垂直分辨率，像素/米。A00F 0000 为 0000 0FA0h＝4000。

(15) 47～50：位图使用的颜色索引数。设为 0，则说明使用所有调色板项。

(16) 51～54：对图像显示有重要影响的颜色索引的数目。若为 0，则表示所有颜色都重要。

2.5.2 JPEG 文件

JPEG 格式是一种常用的图像文件格式，其文件后缀名为.jpeg 或.jpg，它是 Joint Photographic Experts Group(联合图像专家组)的英文缩写。JPEG 格式是一个比较成熟的图像有损压缩格式，能将图像数据压缩到很小的存储空间并保持较好的图像质量。也就是说，虽然图像存储为 JPEG 格式会有一些信息损失，但人眼很难分辨出压缩前后图像质量的差别。JPEG 图像存储格式既满足了视觉系统对色彩和分辨率的要求，又在图像的清晰度和文件大小的选择上找到了一个很好的平衡点。

JPEG 格式是一种灵活的图像文件格式，具有调节图像质量的功能，它允许采用不同的压缩比对图像进行压缩。根据对图像质量的要求，JPEG 格式的压缩比通常为 10∶1～40∶1。压缩比越小，图像质量越高；反之，压缩比越高，图像质量越差。JPEG 文件被压缩的数据主要是高频信息，能较好地保护低频信息，同时也能较好地保护彩色信息，适应于连续色调的图像保存。

随着多媒体技术的应用越来越广泛，传统的 JPEG 压缩算法的缺点和不足越来越明显地表现出来。例如，若被压缩的图像中具有大片近似的色彩信息时，用 JPEG 压缩算法就容易产生马赛克效应，因此，当对图像的质量要求较高时，传统的 JPEG 压缩算法可能达不到要求。为了满足新的应用需求，专家组对 JPEG 格式进行了改进，推出了 JPEG 2000 标准，

新的压缩算法最大的特点是将传统 JPEG 压缩算法中的首选 DCT 算法改为了首选小波变换压缩算法。JPEG 2000 具有以下 4 个特点：

(1) 压缩比高。与传统的 JPEG 压缩算法相比，JPEG 2000 的压缩比提高了 20%～30%。

(2) 同时支持有损压缩和无损压缩，误差稳定性好，图像的质量更好。

(3) 渐进传输。网络下载图像时，JPEG 2000 按块传输，有助于快速浏览网络信息，提高了网络下载的速度。

(4) 支持感兴趣的 ROI，能制定 ROI，可单独对 ROI 进行压缩和解压操作。

JPEG 图像文件结构定义了一些列标记用于区分和识别图像数据及其相关信息，每一个标记由 2 字节组成，并规定前一个字节是 0xFF，每个标记前还可添加若干 0xFF 字节，且数量不限。JPEG 格式的 8 个重要标记如下。

(1) SOI 0xD8：图像开始。

(2) APP0 0xE0：APP0 段主要存储图片的识别信息(字符串“JFIF\0”)，包括 APP0 的长度、标识符、版本号、X 和 Y 的密度单位(units 为 0 表示无单位；units 为 1 表示密度单位为“点数/英寸”；units 为 2 表示密度单位为“点数/cm”；)，X 方向像素密度，Y 方向像素密度，缩略图水平像素数目，缩略图垂直像素数目及缩略图 RGB 位图信息等。

(3) APPn 0xE1～0xEF：其他应用数据块，包括 APPn 的长度等信息。

(4) DQT 0XDB：包括量化表、量化表长度和量化表数目等。颜色可分为 RGB 三个分量，但在 JPEG 文件的 RGB 色彩格式需要先转化为 YUV 的格式。Y 分量代表亮度信息，U、V 分量代表色差信息。通常人眼对于 Y 分量更为敏感，量化表正是基于这一原理进行信息压缩的。

(5) SOF0 0xC0：图像大小、每个像素的位数及 YUV 每个分量的采样信息等。即图像数据帧的开始，包括帧开始长度与精度、每个颜色分量、像素位数、图像高度与宽度、颜色分量数、每个颜色分量的 ID、垂直方向的样本因子、水平方向的样本因子、量化表号等。

(6) DHT 0xC4：定义 Huffman 表，包括表的长度、类型、直流或交流量表、索引、位表和值表。

(7) SOS 0xDA：扫描线开始。包括扫描开始长度、颜色分量数、直流系数表号、交流系数表号、缩略图数据等。

(8) EOI 0xD9：图像全部数据结束。

JPEG 文件格式更深入的了解可参考第 9 章中关于 JPEG 标准的相关介绍。

2.5.3 PNG 文件

PNG 图像文件存储格式即可移植网络图形格式(portable network graphic format，PNG)，是一种位图文件(bitmap file)存储格式。PNG 用于存储灰度图像时，灰度图像的深度可达 16 位；存储彩色图像时，彩色图像的深度可达 48 位，并且还可存储多到 16 位的 α 通道数据。PNG 采用从 LZ77 派生的 Zlib 无损数据压缩算法，压缩比高，生成文件容量小，一般应用于 Java 程序或网页。

1. PNG 文件的特点

PNG 文件具有以下特点：

(1) 体积小。网络通信中因带宽制约，在保证图片清晰、逼真的前提下，网页上不可能

大范围地使用文件较大的 bmp 格式文件。

(2) 无损压缩。PNG 文件采用 LZ77 算法的派生算法进行压缩，可获得高的压缩比。其编码方法可标记重复出现的数据，因而对图像的颜色没有影响，也不会产生颜色的损失，因此可以重复保存而不降低图像质量。

(3) 索引彩色模式。PNG-8 格式与 GIF 图像类似，采用 8 位调色板将 RGB 彩色图像转换为索引彩色图像。图像中保存的不再是各个像素的彩色信息，而是从图像中挑选出来的具有代表性的颜色编号，每一编号对应一种颜色，图像的数据量也因此减少，这对彩色图像的传输非常有利。

(4) 优化的网络传输性。PNG 图像在浏览器上采用流式浏览，图像即使经过交错处理，在完成下载之前，可提供一个基本的图像内容，然后再逐渐显示出清晰图像。它允许连续读出和写入图像数据，这个特性很适合在通信过程中显示和生成图像。

(5) 支持透明效果。PNG 可以为原图像定义 256 个透明层次，使得彩色图像的边缘能与任何背景平滑地融合，从而彻底地消除锯齿边缘，这一功能是 GIF 和 JPEG 没有的。

(6) PNG 同时还支持真彩和灰度级图像的 Alpha 通道透明度。

2. PNG 文件结构

PNG 文件整体上由 PNG 文件标志(PNG file signature)和数据块(chunks)两部分构成。即 PNG 图像格式文件(或者称为数据流)由一个 8 字节的 PNG 文件署名域和按照特定结构组织的 3 个以上的数据块组成。文件的数据块部分一般由 3 个以上的 PNG 的数据块按照特定的顺序组成，因此，一个标准的 PNG 文件结构应该如图 2-9 所示。

PNG文件标志	PNG数据块	…	PNG数据块

图 2-9 标准的 PNG 文件结构

1) PNG 文件标志

PNG 文件署名域由 8 字节的数据组成，用于识别该文件是否 PNG 文件。该域的值如表 2-3 所示。

表 2-3 PNG 文件署名域的值

十进制数	137	80	78	71	13	10	26	10
十六进制数	89	50	4E	47	0D	0A	1A	0A

其中第一个字节 0x89 超出了 ASCII 字符的范围，这是为了避免某些软件将 PNG 文件当做文本文件处理。

2) PNG 数据块

PNG 定义了两种类型的数据块：一种称为关键数据块(critical chunk)，这个标准数据块是必需的；另一种称为辅助数据块(ancillary chunks)，是可选数据块。关键数据块定义了 4 个标准数据块，每个 PNG 文件都必须包含它们，PNG 读写软件也需支持这些数据块。虽然 PNG 文件规范没有要求 PNG 编译码器对可选数据块进行编码和译码，但规范提倡支持可选数据块。

表 2-4 所示是 PNG 数据块类别(关键数据块部分用深色背景加以区分)。

表 2-4 PNG 文件格式中的数据块

数据块符号	数据块名称	多数据块	可选否	位置限制
IHDR	文件头数据块	否	否	第一块
cHRM	基色和白色点数据块	否	是	在 PLTE 和 IDAT 之前
gAMA	图像 γ 数据块	否	是	在 PLTE 和 IDAT 之前
sBIT	样本有效位数据块	否	是	在 PLTE 和 IDAT 之前
PLTE	调色板数据块	否	是	在 IDAT 之前
bKGD	背景颜色数据块	否	是	在 PLTE 之后 IDAT 之前
hIST	图像直方图数据块	否	是	在 PLTE 之后 IDAT 之前
tRNS	图像透明数据块	否	是	在 PLTE 之后 IDAT 之前
oFFs	(专用公共数据块)	否	是	在 IDAT 之前
pHYs	物理像素尺寸数据块	否	是	在 IDAT 之前
sCAL	(专用公共数据块)	否	是	在 IDAT 之前
IDAT	图像数据块	是	否	与其他 IDAT 连续
tIME	图像最后修改时间数据块	否	是	无限制
tEXt	文本信息数据块	是	是	无限制
zTXt	压缩文本数据块	是	是	无限制
fRAc	(专用公共数据块)	是	是	无限制
gIFg	(专用公共数据块)	是	是	无限制
gIFt	(专用公共数据块)	是	是	无限制
gIFx	(专用公共数据块)	是	是	无限制
IEND	图像结束数据	否	否	最后一个数据块

为简单和方便,假设在本节所使用的 PNG 图像文件中,这 4 个数据块按表 2-4 所示的先后顺序进行存储,并且只出现一次。

(1) 数据块结构。PNG 文件中,每个数据块由 4 个部分组成,如表 2-5 所示。

表 2-5 PNG 数据块的结构

名　称	字节数	说　明
Length (长度)	4	指定数据块中数据域的长度,长度不超过($2^{31}-1$)字节
Chunk Type Code(数据块类型码)	4	数据块类型码由 ASCII 字母(A～Z 和 a～z)组成
Chunk Data (数据块数据)	可变	存储按照 Chunk Type Code 指定的数据
CRC (循环冗余检测)	4	存储用于检测是否有错误的循环冗余码

CRC(cyclic redundancy check)域中的数据是对 chunk type code 域和 chunk data 域中的数据进行计算得到的,CRC 算法在 ISO 3309 和 ITU-T V.42 中定义。

(2) IHDR。文件头数据块 IHDR(header chunk)包含有 PNG 文件中存储的图像数据的基本信息,并作为第一个数据块出现在 PNG 数据流中,而且一个 PNG 数据流中只能有一个文件头数据块。

文件头数据块由 13 字节组成,它的格式如表 2-6 所示。

表 2-6 文件头数据块的组成

域的名称	字节数	说明
Width	4	图像宽度,以像素为单位
Height	4	图像高度,以像素为单位
Bit depth	1	图像深度： 索引彩色图像：1、2、4 或 8 灰度图像：1、2、4、8 或 16 真彩色图像：8 或 16
ColorType	1	颜色类型： 0：灰度图像，1、2、4、8 或 16 2：真彩色图像,8 或 16 3：索引彩色图像,1、2、4 或 8 4：带 α 通道数据的灰度图像,8 或 16 6：带 α 通道数据的真彩色图像,8 或 16
Compression method	1	压缩方法(LZ77 派生算法)
Filter method	1	滤波器方法
Interlace method	1	隔行扫描方法： 0：非隔行扫描 1：Adam7(由 Adam M. Costello 开发的 7 遍隔行扫描方法)

(3) PLTE。调色板数据块 PLTE(palette chunk)包含与索引彩色图像(indexed-color image)相关的彩色变换数据,它仅与索引彩色图像有关,而且应置于图像数据块(image data chunk)之前。

PLTE 数据块是定义图像的调色板信息,PLTE 可以包含 1～256 个调色板信息,每一个调色板信息由 3 个字节组成,如表 2-7 所示。

表 2-7 调色表信息表

颜色	字节数	意义
Red	1	0=黑色，255=红色
Green	1	0=黑色,255=绿色
Blue	1	0=黑色,255=蓝色

因此,调色板的长度是 3 的倍数,否则是一个非法的调色板。对于索引图像,调色板信息是必须的,调色板的颜色索引从 0 开始编号,然后是 1、2……调色板的颜色数不能超过色深中规定的颜色数(如图像色深为 4,调色板的颜色数为 16),否则将导致 PNG 图像不合法。

真彩色图像和带 α 通道数据的真彩色图像也可以有调色板数据块,目的是便于非真彩色显示程序用它来量化图像数据,从而显示该图像。

(4) IDAT。图像数据块 IDAT(image data chunk)用于存储 PNG 图像的灰度数据,在数据流中可包含多个连续顺序的图像数据块。

(5) IEND。图像结束数据 IEND(image trailer chunk)用于标记 PNG 文件或者数据流已经结束,并且必须放在文件的尾部。PNG 文件总是以“00 00 00 00 49 45 4E 44 AE 42 60

82"12 个字符结束。这是因为根据数据块结构的定义，IEND 数据块的长度总是 0(00 00 00 00)，数据标识为 IEND(49 45 4E 44)，因此，CRC 码是 AE 42 60 82。

2.5.4　TIFF 文件

TIFF 即标签图像文件格式(tagged image file format)，以.tif 为扩展名，是一种主要用于存储包括照片和艺术图片在内的图像的文件格式。它最初由 Aldus 和微软一起为 PostScript 打印而开发。TIFF 格式支持 256 色、24 位真彩色、32 位、48 位等多种色彩位，同时支持 RGB、CMYK 及 YCbCr 等多种彩色模式，并支持多平台。TIFF 文件可以不压缩(文件较大)也可以压缩，支持 RAW、RLE、LZW、JPEG、CCITT 3 组和 4 组等多种压缩方式，用 Photoshop 编辑的 TIFF 文件可以保存路径和图层，绝大多数图像系统都支持这种格式。

TIFF 在业界得到了广泛的支持，如 Adobe 公司的 Photoshop、Jasc 的 GIMP、Ulead PhotoImpact 和 Paint Shop Pro 等图像处理应用、QuarkXPress 和 Adobe InDesign 桌面印刷和页面排版应用，扫描、传真、文字处理、光学字符识别和一些其他应用等都支持该格式。TIFF 与 JPEG 和 PNG 一起成为流行的高位彩色图像格式，TIFF 文件是最复杂的一种位图文件格式，它广泛应用于对图像质量要求较高的图像存储与转换。由于 TIFF 格式结构灵活、包容性大，因此已成为图像文件格式的一种标准。

TIFF 最初的目的是为 20 世纪 80 年代中期桌面扫描仪厂商避免每商家都使用自己的格式而达成的一个公用统一的扫描图像文件格式。当时的桌面扫描仪只能处理二值图像，因此初期 TIFF 也只是二值图像格式，随着扫描仪的功能越来越强大，TIFF 逐渐支持灰度图像和彩色图像。

1. TIFF 文件结构

TIFF 文件数据格式由图像文件头、图像文件目录、目录入口和图像数据 4 个部分组成，如图 2-10 所示。

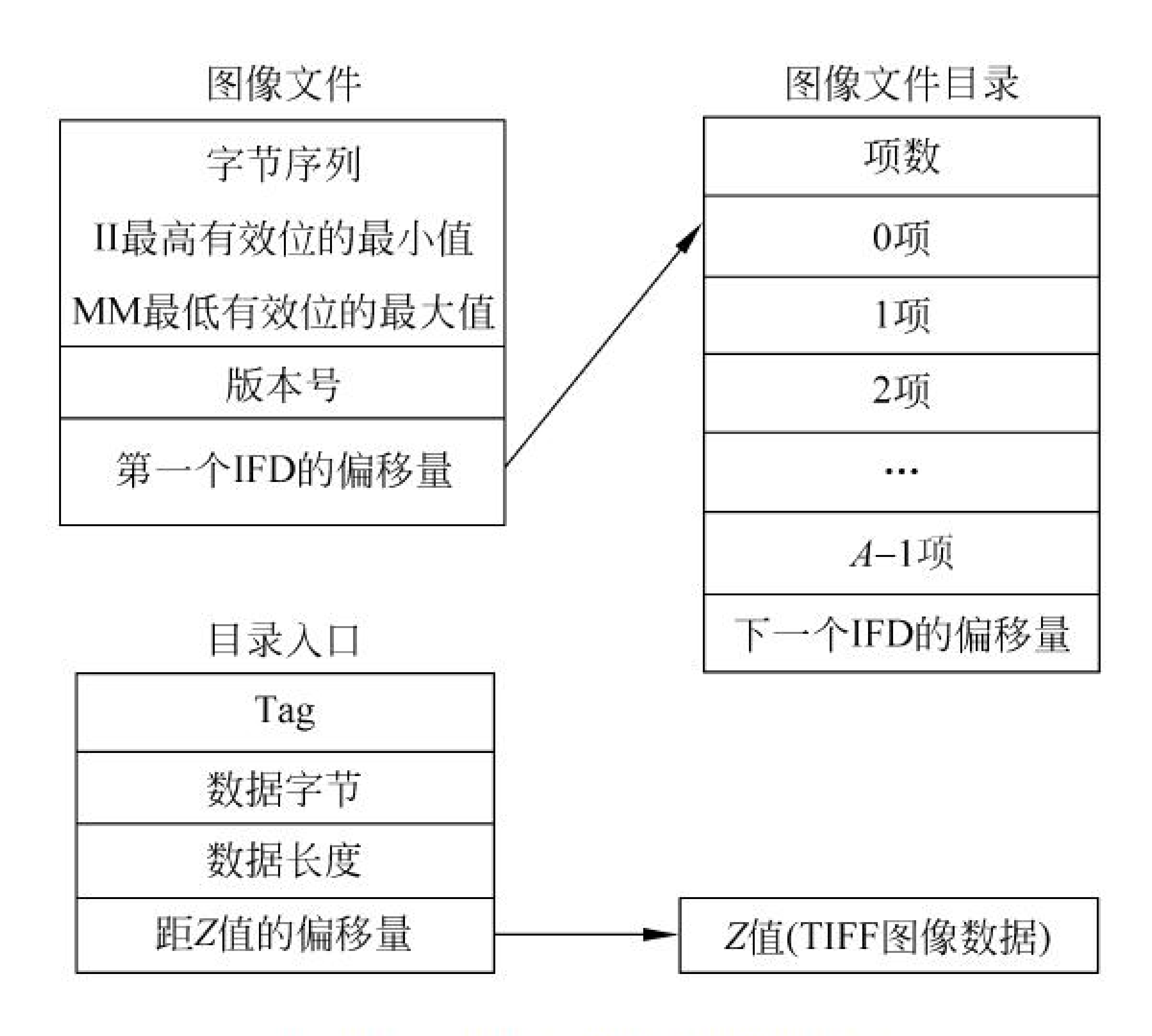


图 2-10　TIFF 文件格式的结构

TIFF 文件的应用非常广泛，这是因为：

(1) TIFF 可以描述多种类型的图像。

(2) TIFF 拥有一系列的压缩方案可供选择。

(3) TIFF 不依赖于具体的硬件。

(4) TIFF 是一种可移植的文件格式。

(5) TIFF 是一种主要的压缩方法。

(6) 多种色彩表示方法。

2. 文件头

每一个 TIFF 文件的第一个数据结构称为图像文件头(image file header,IFH)，它是图像文件体系结构的最高层。文件头部分共 8 字节，格式如图 2-11 所示。

II/MM (表示字节序，共2字节)	42 (版本号，共2字节)
文件目录在整个文件的偏移量，是一个指针，指向IFD的开始部分 (4字节)	

图 2-11　文件头格式

这个结构在一个 TIFF 文件中是唯一的，有固定的位置，位于文件的开始部分，包含了正确解释 TIFF 文件其他部分所需的必要信息。

3. 文件目录

文件目录(image file directory,IFD)是 TIFF 文件的第 2 个数据结构，包含了文件的重要信息，即 IFD 提供了一系列的指针，这些指针包含了各种有关的数据字段在文件中的开始位置，并给出每个字段的数据类型及长度。其格式如图 2-12 所示。

一个 TIFF 图像可有多个 IFD，说明该文件有多幅图像。每一个标记一幅图像的基本属性。IFD 结构包含 3 个成员，其中项数(directory entry count)占 2 字节，指出该结构有多少个目录入口；随后是线性排列的 Directory Entry 序列，数量不定，这就是 TIFF 文件称为可扩充标记的文件的原因，用户甚至可以添加自定义的标记属性，每一个 directory entry 标记了图像的某一属性；最后一个成员是偏移量，标记下一个文件目录相对于文件开始处的位置。

IFD的总项数（2字节）	项1(12字节)
项2 … (12字节)	
下一个IFD的开始地址(用于保存多个文件)	

图 2-12　文件目录格式

4. 目录入口

每个目录入口(directory entry,DE)共由 12 字节组成，格式如图 2-13 所示。

Tag (2字节)	Tag的数据类型
数据长度	
数据值或值的偏移(文件范围)	

图 2-13　目录入口格式

一个 DE 就是一幅图像的某一属性，它包含图像的大小、分辨率、压缩信息、像素的行与列数量、每一像素由几位表示（1 表示二值图像，8 代表 256 色）等。目录入口中的 Tag 成员是该属性的编号，编号按升序排列。第 2 个成员是数据类型，共有 5 种数据类型。第 3 个成员是数据的长度，第 4 个成员是到 Z 值的偏移量，是指 Tag 标识的代表的变量值相对于文件开始处的偏移量，如果该值小于 4 字节，则直接将该值存放即可。

5. 图像数据

根据 IFD 所指向的地址，紧随上述数据之后就是图像数据；图像数据的存储形式有不同的方式。TIFF-B 用于二值图像，TIFF-G 用于灰度图像，TIFF-P 用于带调色板的图像，TIFF-R 用于 RGB 真彩色图像。

TIFF 文件格式适于在应用程序之间和计算机平台之间的文件交换，它的出现使得图像数据交换变得简单。TIFF 是一个灵活适应性强的文件格式，通过在文件头中包含"标签"它能够在一个文件中处理多幅图像和数据。标签能够标明图像的大小或定义图像数据如何排列及是否使用压缩选项。如 TIFF 可以包含 JPEG 和行程长度编码压缩的图像。TIFF 文件也可包含基于矢量的裁剪区域（剪切或者构成主体图像的轮廓）。使用无损格式存储图像的能力使 TIFF 文件成为图像存档的有效方法。与 JPEG 不同，TIFF 文件可以编辑然后重新存储而不会有压缩损失。

每个 TIFF 文件都是从指示字节顺序的两个字节开始的。II 表示小字节在先，MM 表示大字节在先。后面两个字节表示数字 42。42 的读取方法取决于头两个字节所表示的字节顺序。整个文件根据所指出的字节顺序进行读取。

TIFF 的最大局限在于用 4byte 表示偏移量，这样导致文件大小的上限为 4GB。在 20 年前制定 TIFF 标准时 4GB 足够大，但现在却成了制约 TIFF 的一个瓶颈。目前 BigTIFF 已经提出用 8 个字节来表示偏移量。这样数据量应该足够大了。此外，TIFF 文件的复杂性给它的应用带来了一些问题。

2.5.5　GIF 文件

GIF（graphics interchange format）即图像互换格式，是 CompuServe 于 1987 年开发出来的图像文件格式。GIF 文件的数据采用了一种基于 LZW 算法的连续色调的无损压缩格式，允许用户为图像设置背景的透明属性，其压缩率一般在 50%左右，它不属于任何应用程序。目前几乎所有相关软件都支持 GIF，GIF 图像文件是一种在网络上非常流行的图形文件格式。GIF 图像文件格式的特点之一是数据是经过压缩的，其采用的是可变长度压缩算法。GIF 格式的另一个特点是其在一个 GIF 文件中可以存多幅彩色图像，若将存储在一个文件中的多幅图像数据逐幅读出并显示到屏幕上，就可构成一种最简单的动画。

GIF 格式由 CompuServe 引入应用后，因其体积小和成像相对清晰，在早期的互联网上大受欢迎。GIF 文件采用无损压缩技术，对于 256 色以内的图像，则可既减少文件的大小，又保持图像的质量。GIF 图像是基于颜色列表的（存储的数据是该点颜色对应于颜色表的索引值），最多支持 8 位（256 色）。GIF 文件内部分成许多存储块，用于存储多幅图像或者是决定图像表现行为的控制块，用以实现动画和交互式应用。

GIF 主要分为两个版本，即 GIF 89a 和 GIF 87a。

（1）GIF87a：1987 年制定的版本。

(2) GIF89a：1989年制定的版本。在该版本中，为GIF文档扩充了图形控制区块、备注、说明、应用程序编程接口等四个区块，并提供了对透明色和多帧动画的支持。

GIF文件内部是按块划分的，包括控制块和数据块两种。控制块控制数据块的行为，不同的控制块包含不同的控制参数。数据块只包含一些8bit的字符流，由它前面的控制块来决定它的功能，数据块长度为0～255字节，数据块的第一个字节是该数据块的长度，计算数据块的长度时不包括这个字节，所以一个空的数据块也有一个字节，那就是数据块的大小&H00。控制块中的逻辑屏幕描述块和全局彩色表的作用范围是整个数据流，其他控制块仅控制跟在它们后面的图形描述块。一个GIF文件的结构可分为文件头(file hearder)、GIF数据流(GIF data stream)和文件结束块(trailer)三个部分。GIF文件的典型结构如表2-8所示。

表 2-8 GIF 文件的基本结构

序号	结构名称	中文含义	长度/字节
1	Header	GIF文件头	6
2	Logical Screen Descriptor	逻辑屏幕描述块	7
3	Global Color Table	全局彩色表	≤768
		扩展模块(任选)	
4	Image Descriptor	图像描述块	10
5	Local Color Table	局部彩色表(可重复 n 次)	≤768
6	Table Based Image Data	表基图像数据	
7	Graphic Control Extension	图形控制扩展块	8
8	Plain Text Extension	无格式文本扩展块	
9	Comment Extension	注释扩展块	4～258
10	Application Extension	应用程序扩展块	
		扩展模块(任选)	
11	GIF Trailer	文件结束块	1

根据表2-8所示的基本结构，对GIF文件详述如下。

1. GIF 文件头

GIF文件头描述块定义GIF数据流，文件头由GIF标记域(signature)和版本号(version)域组成，是一个由6个固定字节组成的数据块，它们用来说明使用的文件是GIF格式及当前的版本号，标志域和版本域各占3字节。GIF标记域存放的是GIF，版本号域存放的是1987年5月发布的87a或1989年7月发布的89a，或更新的版本号。

GIFHEADER结构定义如下：

```
typedef struct gifheader
{
    BYTE bySignature[3];
    BYTE byVersion[3];
}GIFHEADER;
```

2. 逻辑屏幕描述块

逻辑屏幕描述块(logical screen descriptor)包含定义图像显示区域的参数，包括背景颜色信息。这个数据块中的坐标是相对于虚拟屏幕的左上角开始的，不一定是指显示屏的绝

对坐标，因此可以参照窗口软件环境下的窗口坐标或者打印机坐标来设计图像显示程序。逻辑屏幕描述块的结构如表 2-9 所示。

表 2-9 逻辑屏幕描述块的结构

7	6	5	4	3	2	1	0	字 节 号	域 的 名 称	类 型
Logical Screen								0	逻辑屏幕宽度	unsigned
								1	以像素为单位	
Logical Screen								2	逻辑屏幕高度	unsigned
								3	以像素为单位	
G	CR			S	Size			4	包装域	
Background Color Index								5	背景颜色索引	Byte
Pixel Aspect Ratio								6	像素宽高比	Byte

GIFSCRDESC 结构定义如下：

```
typedef struct gifscrdesc
{
    WORD wWidth;
    WORD wDepth;
    struct globalflag
    {
     BYTE PalBits: 3;
     BYTE SortFlag: 1;
     BYTE ColorRes: 3;
     BYTE GlobalPal: 1;
    }
    GlobalFlag;
    BYTE byBackground;
      BYTE byAspect;
}GIFSCRDESC;
```

逻辑屏幕描述块包含 7 字节。字节 0 和字节 1 用于说明逻辑显示屏的宽度，字节 2 和字节 3 用于说明逻辑显示屏的高度，字节 4 用于描述彩色表的属性，字节 5 用于指定背景颜色索引，字节 6 用于计算像素的宽高比。相关细节详述如下。

(1) 屏幕描述块中的第 5 个字节称为包装域(packed fields)，它由 4 个子域组成：

① 全局彩色表标志(global color table flag)域 G 用来说明是否有全局彩色表存在。若 G=1，表示有一个全局彩色表将紧跟在这个逻辑屏幕描述块之后；这个标志也用于选择背景颜色索引(background color index)。如果 G=1，背景颜色索引域中的值就用作背景颜色的索引。

② 彩色分辨率(color resolution)域 CR 用于表示原始图像可用的每种基色的位数(实际值减 1)。这个位数表示整个调色板的大小，而不是这幅图像使用的实际颜色数。若该域的值 CR=3，说明原始图像可用每个基色有 4 位的调色板来生成彩色图像。

③ 彩色表排序标志(sort flag)域 S 用于表示全局彩色表中的颜色是否按重要性(或使用率)排序。若 S=0，表示没有重要性排序；若 S=1，表示最重要的颜色排在前。这样做的目的是辅助颜色数比较少的解码器能够选择最好的颜色子集，在这种情况下解码器就可选

择彩色表中开始段的彩色来显示图像。

④ 全局彩色表大小(size of global color table)域 Size 表示每个像素的位数,它用于计算全局彩色表中包含的字节数。在全局彩色表标志域 G=0 时就不用计算,G=1 时就需计算彩色表的大小,具体计算参考本节全局彩色表。

(2) 屏幕描述块中的第 6 个字节是背景颜色索引,它是彩色表的一个索引值,用于指定背景颜色。若全局彩色表标志域 G=0,这个域的值也设置为 0。

(3) 像素宽高比(pixel aspect ratio)域中的值是一个因数,是计算原始图像像素的宽高比的一个近似值。该域的值范围为 1~255。若该值不等于 0,则宽高比的近似值按下式计算:

$$\text{Aspect Ratio}=(\text{Pixel Aspect Ratio}+15)/64$$

像素宽高比定义成像素的宽度与高度之比,比值的范围在 4∶1~1∶4 之间,其增量为 1/64。

3. 全局彩色表

由于一个 GIF 文件可以包含多幅彩色图像,每幅彩色图像也许都包含适合自身特点的彩色表,因此一个 GIF 文件可以有好几个彩色表。但归纳起来只有两类:全局彩色表和局部彩色表。全局彩色表可用于图像本身没有带彩色表的所有图像和无格式文本扩展块(plain text extension)图像,而局部彩色表只用于紧跟在它后面的一幅图像。在处理全局彩色表和局部彩色表时需要注意如下规则。

(1) 如果 GIF 文件包含全局彩色表,而且要显示的图像本身又带有局部彩色表,则显示该幅彩色图像时就用自有的彩色表,而不用全局彩色表。在这种情况下,解码器就首先保存全局彩色表,然后使用局部彩色表显示图像,最后再恢复全局彩色表。

(2) 全局彩色表和局部彩色表都是可选的。基于此,解码器最好要保存全局彩色表直到出现另一个全局彩色表为止。这样,对于包含完全没有彩色表的一幅或者多幅彩色图像的 GIF 文件就可以使用最后保存的全局彩色表进行处理。

(3) 如果同类型的图像能够使用相同的彩色表来显示,编码器就要尽可能使用一个全局彩色表;如果没有彩色表可用,解码器就可以使用计算机系统提供的彩色表或者解码器自身的彩色表。

(4) 全局彩色表是否存在由逻辑屏幕描述块中第 5 字节的全局彩色表标志域 G 的值确定,若存在,彩色表就紧跟在逻辑屏幕描述块之后。

4. 图像描述块

GIF 图像文件格式可包含多幅图像,而且无固定存放顺序,仅用一个字节图像分隔符(image separator)来判断是否为图像描述块。每一幅图像都由一个图像描述块、局部彩色表(该表可以没有)和图像数据组成。每幅图像必须在逻辑屏幕描述块中定义的逻辑屏尺寸范围内。图像描述块之前可以有一个或者多个控制块,如图形控制扩展块之后可有一个局部彩色表。无论前后是否有各种数据块,图像描述块总是带有图像数据。图像描述块的结构如下。

```
typedef struct gifimage
{
    WORD wLeft;
```

```
    WORD wTop;
    WORD wWidth;
    WORD wDepth;
    struct localflag
      {
        BYTE PalBits : 3;
        BYTE Reserved : 2;
        BYTE SortFlag : 1;
        BYTE Interlace:1;
        BYTE LocalPal : 1;
      } LocalFlag;
} GIFIMAGE;
```

图像描述块的第 9 个字节称为包装域(packed fields)字节,它由 5 个子域组成：

(1) 局部彩色表标志域 L 用于说明是否有局部彩色表存在。如果 L=1,表示有一个局部彩色表将紧跟在这个图像描述块之后；如果 G=0,表示图像描述块后面没有局部彩色表,该图像将使用全局彩色表。

(2) 交插显示标志(interlace flag)域 I 用于表示该图像是不是交插图像(interlaced images)。如果 I=0,表示该图像不是交插图像；如果 I=1,表示该图像是交插图像。使用该位标志可知道图像数据是如何存放的。GIF 文件格式定义了两种数据存储方式：一种是按图像行连续顺序存储,这个顺序与显示器上显示行的顺序相同；另一种按交插方式存储。交插图像按行分成如下所示的 4 组：

① Group 1：每隔 8 行组成一组,从第 0 行开始显示/第 1 遍交插。

② Group 2：每隔 8 行组成一组,从第 4 行开始显示/第 2 遍交插。

③ Group 3：每隔 4 行组成一组,从第 2 行开始显示/第 3 遍交插。

④ Group 4：每隔 2 行组成一组,从第 1 行开始显示/第 4 遍交插。

显示图像需要较长的时间,使用这种方法存放和显示图像数据可以在图像显示完成之前看到这幅图像的概貌,而不觉得显示时间长。

(3) 彩色表排序标志域的含义与全局彩色表中域的含义相同。

(4) 保留(reserved)。

(5) 局部彩色表大小域的值用于计算局部彩色表中所包含的字节数。

5. 局部彩色表

局部彩色表用于紧跟在它后面的图像。彩色表是否存在取决于图像描述块中局部彩色表标志位的设置,彩色表的结构和大小与全局彩色表完全相同。

6. 表基图像数据

GIF 图像采用了 LZW 算法对图像数据进行压缩。为提高压缩编码的效率,对 LZW 编码器输出的代码采用可变长度码 VLC。表基图像数据(table based image data)由 LZW 最小代码长度和图像数据组成,LZW 最小代码长度域的值用于确定图像数据中 LZW 代码使用的初始位数。图像数据由数据子块(data sub-blocks)序列组成。数据子块的结构是一个可变长度的数据块,其长度由块大小域(block size)域中的值确定,字节数为 0～255。

7. 图形控制扩展块

图形控制扩展块(graphic control extension)包含处理图形描绘块时需使用的参数,它

的结构说明如下：

（1）扩展导入符(extension introducer)用于识别扩展块的开始，域中的值是一个数值等于 0x21 的固定值。

（2）图形控制标签用于标识当前块是一个图形控制扩展块，域中的值是一个数值等于 0xF9 的固定值。

（3）块大小(block size)用于说明该扩展块所包含字节数，该字节数是从这个块大小域之后到块结束符之间的字节数。控制块数据结果如下：

```
typedef struct gifcontrol
{
    BYTE byBlockSize;
      struct flag
      {
        BYTE Transparency :1;
        BYTE UserInput     :1;
        BYTE DisposalMethod:3;
        BYTE Reserved       :3;
      }Flag;
      WORD wDelayTime;
      BYTE byTransparencyIndex;
      BYTE byTerminator;
} GIFCONTROL;
```

（4）处理方法(disposal method)规定图形显示之后译码器要用规定的方法进行处理。用户输入标志(user input flag)域表示在继续处理之前是否需要用户输入响应。在延时时间和用户输入标志都设置为 1 的情况下，继续处理的开始时间取决于用户响应输入在前还是延时时间结束在前。

（5）透明(transparency flag)表示是否给出透明索引。

（6）延时时间(delay time)用于指定在图形显示之后继续处理数据流之前的等待时间，一百分之一秒为单位。

（7）当且仅当透明标志位设置为 1 时，透明索引用于指示处理程序是否要修改显示设备上的相应像点。当且仅当透明标志位设置为 1 时，就要修改。

（8）块结束符(block terminator)表示该图形控制扩展块结束，它是由一个字节组成的数据块，该域的值是一个固定的值 0x00。

8. 无格式文本扩展块

无格式文本扩展块即图像说明扩充块，无格式文本扩展块包含文本数据和描绘文本所需的参数。文本数据用 7 位的 ASCII 字符编码并以图形形式显示。扩展块的结构如下。

```
typedef struct gifplaintext
{
   BYTE byBlockSize;
   WORD wTextGridLeft;
   WORD wTextGridTop;
   WORD wTextGridWidth;
   WORD wTextGridDepth;
```

```
    BYTE byCharCellWidth;
    BYTE byCharCellDepth;
    BYTE byForeColorIndex;
    BYTE byBackColorIndex;
} GIFPLAINTEXT;
```

其中,byBlockSize 用于指定该图像扩充块的长度,取值为 13;wTextGridLeft 用于指定文字显示方格相对于逻辑屏幕左上角的 X 坐标(像素);wTextGridTop 用于指定文字显示方格相对于逻辑屏幕左上角的 Y 坐标;wTextGridWidth 用于指定文字显示方格的宽度;wTextGridDepth 用于指定文字显示方格的高度;byCharCellWidth 用于指定字符的宽度;byCharCellDepth 用于指定字符的高度;byForeColorIndex 用于指定字符的前景色;byBackColorIndex 用于指定字符的背景色。

9. 注释扩展块

注释扩展块(comment extension)域的内容用于说明图形、作者或者其他任何非图形数据和控制信息的文本信息。其中的注释数据是序列数据子块,每块字节数为 1～255。

10. 应用程序扩展块

应用程序扩展块(application extension)包含制作该图像文件的应用程序的相关信息,GIF 中用识别码 0xFF 来判断一个扩充块是否为应用程序扩充块。它的结构定义如下:

```
typedef struct gifapplication
 {
    BYTE byBlockSize;
    BYTE byIdentifier[8];
    BYTE byAuthentication[3];
    } GIFAPPLICATION;
```

其中,byBlockSize 用于指定该应用程序扩充块的长度,其值为 12;byIdentifier 用于指定应用程序名称;byAuthentication 用于指定应用程序的识别码。

11. GIF 文件结束块

结束块(GIF trailer)表示 GIF 文件的结尾,它包含一个固定的数值 0x3B。

2.6　数字图像的显示特性

一般而言,数字图像处理的结果依然是图像,还需要将图像显示或打印以给人们观看或保存。因此可以说,图像的显示是数字图像处理的一个重要步骤。

2.6.1　图像的显示

图像需通过屏幕显示或打印输出才可以观察到,图像屏幕显示的效果由图像的大小、光度分辨率、低频响应、高频响应、点间距和噪声特性等因素决定。

图像显示系统用于显示显示器的物理尺寸和系统可处理的数字图像大小。显示器自身的物理尺寸显然决定了可显示图像的大小,因此它应该尽可能大,以便于观察和理解所显示的图像。显示系统能处理的最大数字图像尺寸是图像处理系统的主要指标之一,显示器尺寸不足会降低整个图像处理设备的效果,显示器必须与待处理和显示的最大图像的行数和

每行像素数相适应。数字图像处理系统的常见显示设备主要包括可随机存取的 CRT(阴极射线管)、电视显示器和 LCD(液晶显示器)。

对于 CRT,电子枪束的水平位置是计算机可以控制的,在每个偏转位置,电子枪束的强度是通过电压进行调制的,而这一电压值与该点对应的灰度值成正比。这样灰度值的变化反映在 CRT 上就是亮度的变化。对于普通的电视监视器若能提供合适的视频信号,也可用作数字图像显示器。

LCD 是另一种常用的显示设备,它是一种低电压、低功耗器件,可直接由 MOS-IC 驱动,因此器件和驱动系统之间的配合较好。LCD 的优点是结构简单、平面型,显示面可任意加工,而且 LCD 是反射型的,在室内条件也容易看到数字图像。目前,台式计算机、笔记本电脑、手表、玩具、计算器、民航候机牌、火车候车室、户外广告,以及测量仪器和航天测控显示等都广泛使用 LCD 液晶显示设备。

计算机系统中,数字图像是通过光栅扫描方式和计算机内的帧缓冲存储器进行屏幕显示的。

首先介绍什么是帧缓冲存储器。任何一台具有显示功能的计算机内都设有专门用于存储图像信息的帧缓冲存储器,计算机时刻自动监测该存储器,如果帧缓冲存储器内被写入图像数据,则这些数据会自动经光栅扫描方式映射到计算机显示屏幕上,从而形成数字图像。帧缓冲存储器中的每一位对应于屏幕上的一个点,当某一位上的数据被置为 1 时,与该数据位对应的计算机屏幕上的相应坐标位置上就出现一个亮点,而数据位为 0 时,屏幕上的相应坐标位置则产生一个暗点。计算机开始启动时,帧缓冲存储器上所有的位被自动置为 0,当读入图像数据后,帧缓冲存储器才会根据图像数据的每一位分别置 0 或置 1。

例如,若一台计算机的显示器分辨率设定为 640×480,为显示一幅二值图像需 640×480 位的帧缓存容量,这个容量称为一个位平面。因此,若显示一幅 256 灰度级的单色图像则需要配置 8 个位平面,即需要 640×480 字节的帧缓存容量。若显示器设置为 800×600、1024×768 或 1280×1024 等高分辨率时,则所需的帧缓存容量要求相应增大。若需要显示 R、G、B 均为 256 级的真彩色图像,则需要进行 R、G、B 三色合成,因此帧缓存的容量是单色 256 灰度级显示容量的 3 倍。

若操作系统允许用户对显示器的分辨率进行设置,则应充分考虑计算机现有的帧缓存容量的容许范围。例如,当现有的帧缓存为 512KB 时,为显示 256 灰度的单色数字图像,最多只能选用 800×600 的分辨率。如果选用 1024×768 的分辨率,则图像的灰度将降至 16 个等级,即 4bit 图像。可以通过增设帧缓存的容量来获取较大的分辨率,需要注意的是,若显示器不具备显示较高分辨率的能力,即使配置了足够的帧缓存,也仍然不能得到高的分辨率。

熟悉帧缓冲存储器的原理和作用之后,则可以通过直接向帧缓存填写图像数据来显示图像。需指出的是,每一个像素上的数据在帧缓存上是以位为单位描述的,而计算机的数据输入/输出一般以字节为单位,因此图像数据中的每一个字节对应显示画面上横向排列的 8 个像素。

2.6.2 图像的打印

数字图像除可进行显示外,还可以通过各种打印设备将图像打印出来。在有些图像处理著作中,将图像打印输出又称为图像的永久显示,将图像永久记录在纸或胶片等介质上都

称为图像的永久显示。图像的打印设备包括图像记录器、打印机和其他图像硬拷贝设备等。

1. 半调输出技术

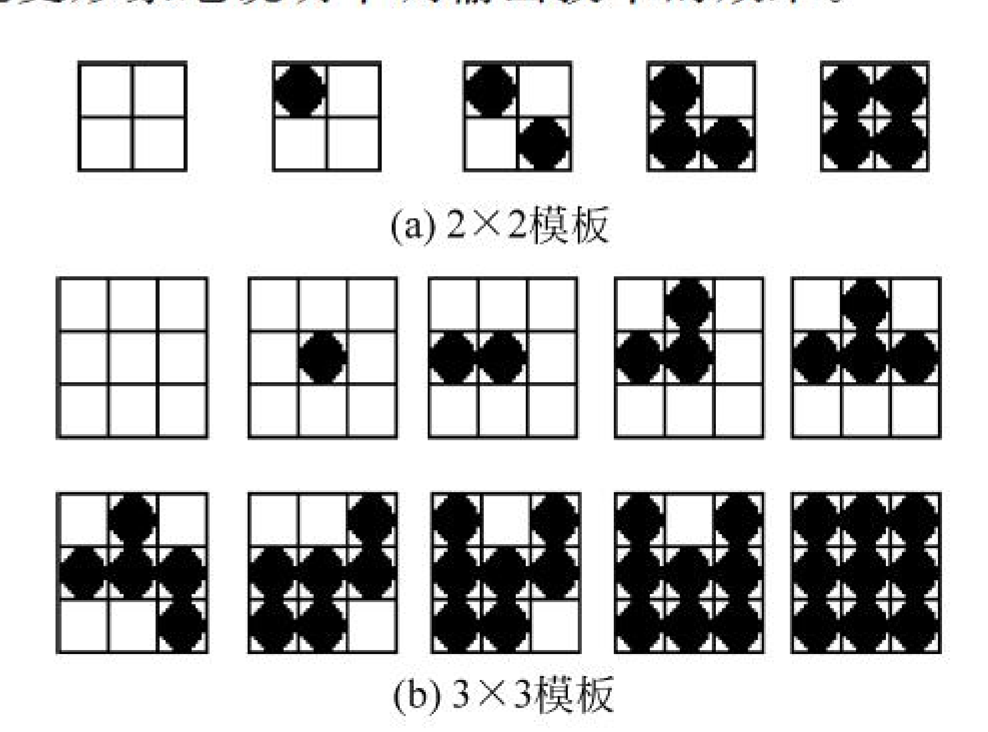
一般情况下，打印设备只能输出二值图像，以激光打印机为例，其输出的灰度只有两极，或者输出为黑色的打印点，或者输出为白色点（无墨）。为了在现行的打印设备上输出灰度图像，常采用半调输出技术。半调输出技术是一种利用人类的视觉原理，将灰度图像转换为二值图像的技术。半调输出的基本原理是将输出图像中的灰度转换为二值点的模式，从而使现有的仅能输出二值图像的打印设备输出灰度图像。

半调输出本质上是一种利用人类视觉原理的特定输出方法，即通过控制二值点输出模式的形式（包括数量、尺寸和形状等）使人类视觉系统获得灰度级别上的视觉效果。也就是说，半调输出利用了视觉系统在微观上的视觉平均特性，以单位面积上二值墨点的多少、墨点的大小来体现不同的灰度等级，如果对半调输出的图像进行足够的放大，呈现出来的依然是二值图像。

2. 半调输出模板

半调输出模板是半调输出的一种具体实现方法，是指将图像输出的单元进行细分，通过邻近的二值点结合起来组成图像输出单元。这样细分之后，图像输出的基本单元包含若干二值点，这些二值点组成一个模板，使模板中的一些二值点输出墨点，而其他二值点的输出为没有墨色的空白点，从而达到显示不同灰度等级的效果。

例如，图 2-14(a)所示为 2×2 模板，即一个输出单元分为 2×2 网格，这样就可以实现 5 个灰度等级的打印输出效果；图 2-14(b)所示为 3×3 模板，可实现 10 个灰度等级效果。需要说明的是，这里将输出的二值点进行了放大，实际打印机的二值点比图 2-14 所示的二值点小很多，放大的目的是更形象地说明半调输出技术的效果。



(a) 2×2模板

(b) 3×3模板

图 2-14 半调输出技术模板

3. 半调输出的种类

半调输出技术主要包括幅度调制技术和频率调制技术两种方法。

1）幅度调制

半调输出技术的早期主要通过控制输出墨点的尺寸来形成灰度图像的显示效果，这就是半调输出技术中的幅度调制（AM）技术。在幅度调制技术中，二值墨点是按规则进行排列的，二值墨点的大小需根据所表示的灰度值而变化，但对二值墨点的形状没有具体要求。以激光打印机为例，其不同的灰度显示效果是通过控制墨水墨点的比例而达到的。实际上，

二值墨点的输出模式不仅取决于每个点的尺寸，也取决于网格的间隔距离大小，间隔越大，分辨率越低，间隔越小，输出的分辨率越高。网格的间隔大小与打印机分辨率（每英寸的打印点数）有关。

2）频率调制

频率调制技术（FM）是相对于幅度调制而言的，所谓频率调制的半调输出是指输出二值点的尺寸是不变的，二值点在空间的分布与灰度有关。二值点的空间分布是指墨点的间隔或者一定区域内墨点出现的频率。显然，当二值点分布比较密集时，将实现较暗的灰度级，二值点分布比较稀疏时，就可以实现较亮的灰度等级。

随着对打印质量要求的不断提高，单一的幅度调制技术和频率调制技术都达到了自身的极限，目前，业界开始采用将幅度调制和频率调制技术相结合的图像输出方法。需要指出的是，对于半调输出技术而言，若需要输出 256 灰度级效果，需采用 16×16 的模板，这样输出图像的空间分辨率将受到较大的影响。因此，半调输出技术仅在输出设备自身的灰度值有限的情况下才可以考虑使用。

2.7 图像质量评价

图像是人们获取信息的重要途径，其所承载的信息非常丰富。图像在获取、处理、传输和存储的过程中，由于各种因素的影响，将可能影响到图像质量，这给图像的后期处理带来了一定的困难。因此，建立图像质量评价机制具有重要的意义，是图像处理工程的基础技术之一。

2.7.1 图像质量评价方法

图像质量评价涉及图像处理的许多方面，如压缩、传输、存储、增强、水印等。一个有效的评价标准至少具有三种应用：第一，可以在质量控制系统中检测图像质量，例如图像采集系统利用其来自动调整系统参数，从而获得更高质量的图像；第二，用于衡量图像处理系统和算法的有效性；第三，可以嵌入到图像处理系统中用于优化系统和参数设置，在视频通信系统中，质量标准既能辅助编码端的预滤波和比特分配算法的设计，又能辅助解码端的最优重构、误差消除和后滤波算法的设计。

图像质量评价可分为主观评价和客观评价两大类方法。主观评价法主要凭借评价人员的主观感知来评价图像质量；客观评价法依据评价的数学模型给出量化指标，模拟人类视觉系统感知机制衡量图像质量。图像质量评价还有一些其他评价方法，如根据有无参考图像又可以将图像质量评价方法分为有参考评价模型和无参考评价模型。有参考评价模型是指根据一幅参考图像，对经过处理的图像进行评价。在图像复原、图像去模糊化等处理时常采用这种评价方法。无参考评价模型是指没有参考图像，直接根据图像的统计特性或观察者对图像的主观打分进行质量评价。

图像质量的主观评价方法（mean opinion score，MOS）考虑了观察者对图像的理解效果，常用方法包括平均主观分值法（MOS）和差分主观分值法（DMOS）。平均主观分值法是通过不同观测者对于图像质量评价得出的主观分值进行平均来得到归一化的分值，以该分值表示该图像质量。评价标准分为优、良、中、差、劣五等。对应这五个标准有两种类型的分

值,即图像主观绝对分值和图像主观相对分值。主观绝对分值是观测者对于图像本身的主观分值,主观相对分值是观测者对于图像在一组图像中的相对其他图像的分值。由于主观评价方法受到观察者知识背景、观测目的和环境等影响,稳定性和可移植性差,且难以用数学模型进行表达。

客观质量评价是指使用一个或多个图像质量的度量指标,建立与图像质量相关的数学模型,采用计算机自动计算出图像质量,其目标是客观评价结果与人的主观感受相一致。

2.7.2 均方误差

图像质量的均方误差评价(mean square error,MSE)方法是一种常用的图像质量评价方法,它是指被评价图像与参考图像对应位置像素值误差的平方均值误差。假设有一幅参考图像 $f(x,y)$,另有一个受到污染的图像 $g(x,y)$,如欲对图像 $g(x,y)$进行质量评价,其均方误差计算公式为

$$\text{MSE} = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[f(x,y)-g(x,y)] \tag{2-6}$$

根据均方误差的定义,误差值越大,说明图像像素值整体差异大,图像质量越差;反之,均方误差越小,说明图像质量越好。均方误差为 0,则被评价图像与参考图像完全一致。

2.7.3 信噪比与峰值信噪比

图像的信噪比也是常用的图像质量评价指标之一,是参考图像像素值的平方均值与均方误差比值的对数的 10 倍。若一幅参考图像用 $f(x,y)$表示,而受到污染的图像用 $g(x,y)$表示,如欲对图像 $g(x,y)$进行质量评价,其信噪比(signal noise ration,SNR)误差计算公式为

$$\text{SNR} = 10\lg\frac{\frac{1}{MN}\sum_{x,y}f^2(x,y)}{\frac{1}{MN}\sum_{x,y}v^2(x,y)} = 10\lg\frac{\sum_{x,y}f^2(x,y)}{\sum_{x,y}[f(x,y)-g(x,y)]^2} \tag{2-7}$$

由式(2-7)可以看出,在均方误差相同的情况下,对于不同的图像,由于像素值不同,其信噪比很可能不一样。

为了消除图像自身像素值大小对评价指标产生的影响,通常采用峰值信噪比(peak signal noise ration,PSNR),这是一种与信噪比相似的质量度量准则。PSNR 定义为图像所容许的最大像素值平方与均方误差比值的对数的 10 倍。如以 8 位灰度图像为例,由于最大像素值为 255,因此峰值信噪比公式为

$$\text{PSNR} = 10\lg\frac{255^2}{\frac{1}{MN}\sum_{x,y}v^2(x,y)} = 10\lg\frac{255^2}{\text{MSE}} \tag{2-8}$$

2.7.4 结构相似度

在图像质量评价机制中,研究人员发现 MSE、SNR、PSNR 等评价方法有时可能与人的视觉感受出现较大的差异。为此,近年来研究了很多更接近人类视觉特性的客观评价方法。其中得到广泛应用和认可的是 Wang 等研究人员提出的结构相似度(structral

similarity，SSIM）评价法。

结构相似度评价方法考虑了两幅图像的亮度、对比度和结构等因素对相似性的影响。图 2-15 所示是结构相似度算法流程。

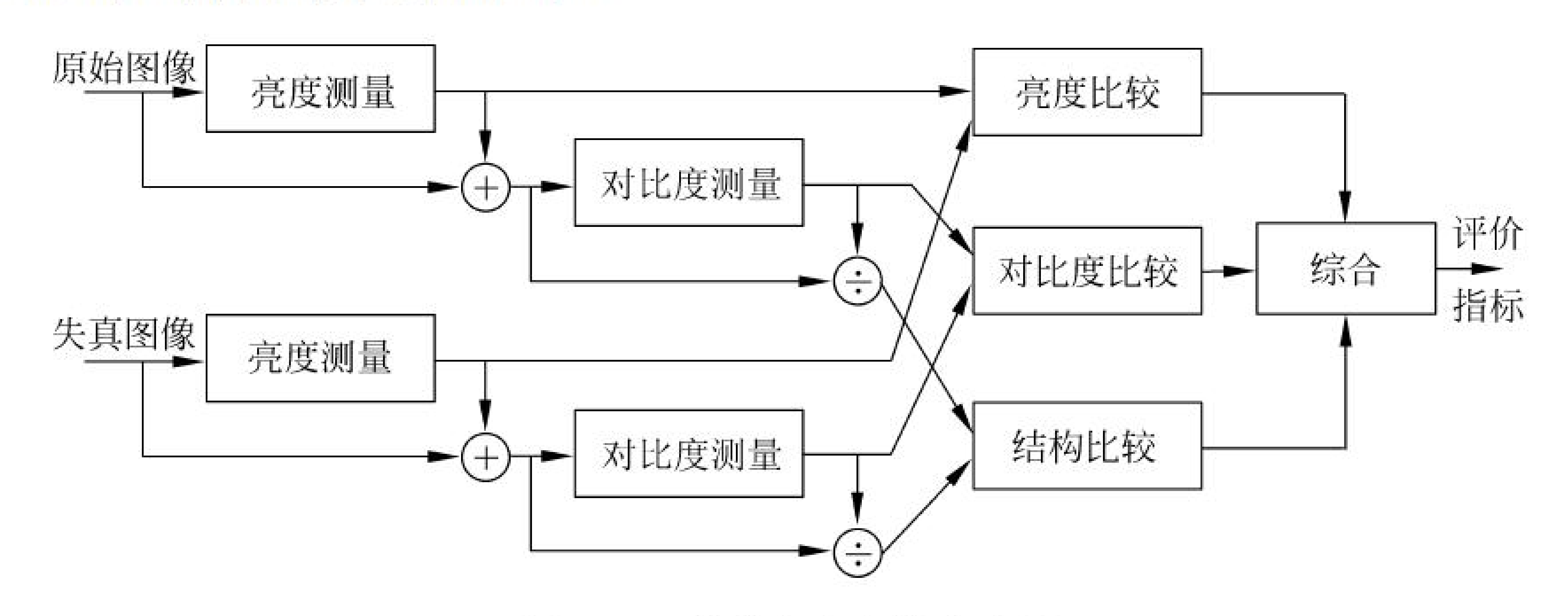


图 2-15 结构相似度算法流程

Wang 提出的结构相似度的计算模型为

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{2-9}$$

一般取 $\alpha=\beta=\gamma=1$，$l(x,y)$、$c(x,y)$、$s(x,y)$分别为亮度相似度、对比度相似度和结构相似度的度量值，为三个正数，用于调节不同因素的影响权重。亮度、对比度和结构相似度分别定义如下：

$$\begin{cases} l(x,y) = \dfrac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \\ c(x,y) = \dfrac{2\sigma_x\mu_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \\ s(x,y) = \dfrac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \end{cases}$$

其中，μ_x，μ_y，σ_x，σ_y，σ_{xy}分别为两幅图像的均值、标准差和协方差；c_1、c_2 和 c_3 为三个远小于最大灰度值平方的常数，通常取值为 $c_1=(k_1L)^2$，$c_2=(k_2L)^2$，$c_3=c_2/2$，k_1、k_2 取远小于 1 的数，L 是指像素的最大值，通常取 $k_1=0.01$，$k_2=0.03$，$L=255$，计算方法如下：

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i,\quad \sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)^2} \tag{2-10}$$

$$\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i,\quad \sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i-\mu_y)^2} \tag{2-11}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y) \tag{2-12}$$

该方法基于光照对于物体结构是独立的，而光照改变主要来源于亮度和对比度，所以它将亮度和对比度从图像的结构信息中分离出来，并结合结构信息对图像质量进行评价。基于这一类原理的方法在一定程度上避开了自然图像内容的复杂性及多通道的去相关问题，直接评价图像的结构相似性。SSIM 的值域范围为[0,1]，并且满足距离度量的三个性质。

SSIM 的值域范围为[0,1]，并且满足距离度量的三个性质。

（1）对称性：$SSIM(x,y)=SSIM(y,x)$。

（2）有界性：$0\leqslant SSIM(x,y)\leqslant 1$。

（3）最大值唯一性：$SSIM(x,y)=1\Leftrightarrow x=y$。

考虑到图像的亮度和对比度与图像内容具有密不可分的关系，无论是亮度还是对比度，在图像的不同位置可能有不同的值，因此，实际应用中通常可将图像分为多个子块，分别计算各个子块的结构相似度，然后由各个子块的相似度计算出平均相似度（mean structure similarity），以该平均值作为两幅图像的结构相似度。

2.8　灰度直方图

直方图是进行数据统计的一个简单、高效和常用的工具，灰度直方图也是数字图像处理中一个方便、有效的常用工具。灰度直方图是基于图像灰度值和像素统计分布的形象表示，它概括地表示了一幅图像的灰度级信息。任何一幅图像的直方图都包括了该图像的许多特征信息，某些特定类型的图像甚至可由直方图来完全描述。

2.8.1　直方图的定义与性质

1. 直方图的定义

直方图是统计学中的常用工具之一，在数字图像处理中灰度直方图是灰度级的函数，它描述了图像中具有该灰度级的像素的个数，其横坐标是灰度级，纵坐标是该灰度出现的频率（像素的个数）。

对于连续函数表示的图像，其灰度直方图也有另一种定义方式。在图 2-16 所示的图像中，有一条灰度级为 D_1 的轮廓线，在更高的灰度级 D_2 处有第二条轮廓线。A_1 表示第一条轮廓线所包围区域的面积，同样，以 A_2 表示第二条轮廓线所包围的区域的面积。

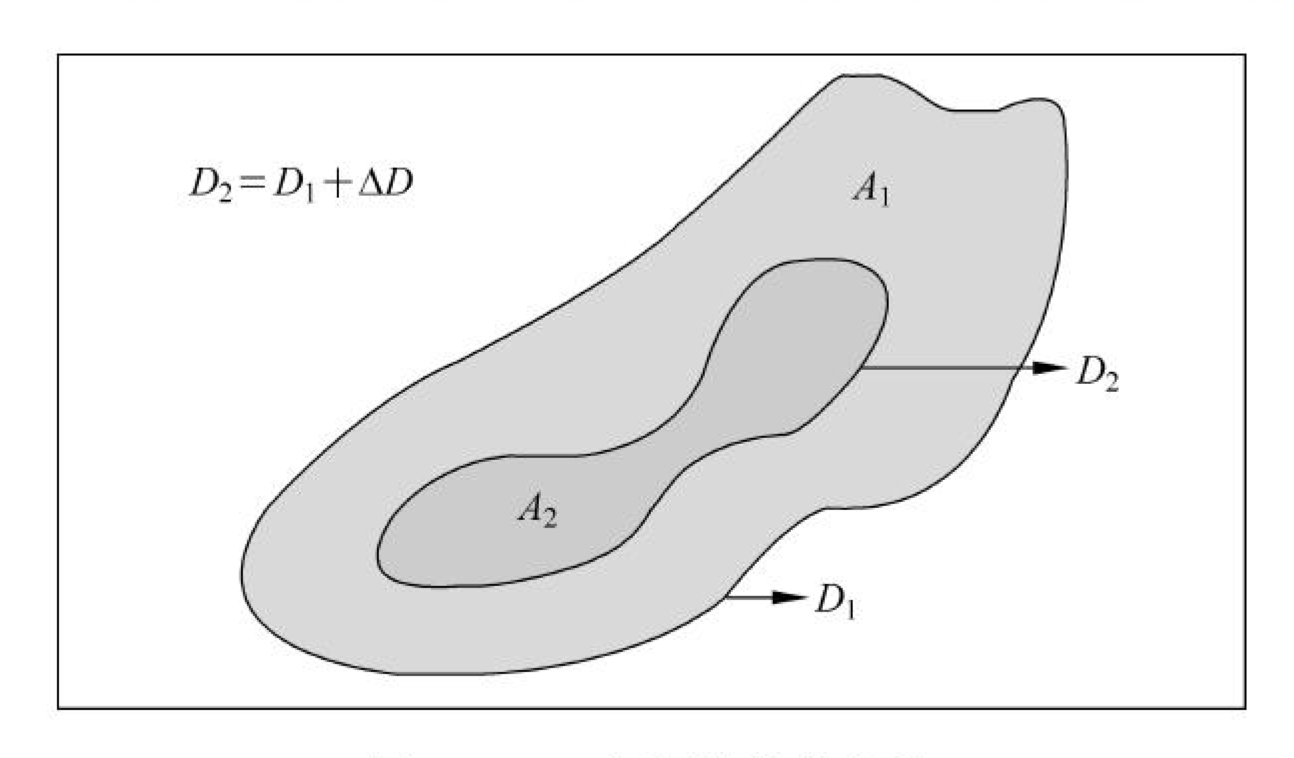


图 2-16　一幅图像的轮廓线

在一幅连续图像中，将灰度级为 D 的轮廓线所包围的面积称为灰度级 D 的阈值面积函数，用 $A(D)$表示，则直方图可定义为

$$H(D)=\lim_{\Delta D\to 0}\frac{A(D)-A(D+\Delta D)}{\Delta D}=-\frac{\mathrm{d}}{\mathrm{d}D}A(D) \tag{2-13}$$

因此，一幅连续图像的直方图是其面积函数的导数的负值。负号的产生是因为随着 D 的增加面积函数 $A(D)$的值在减少。若将图像看成是一个二维的随机变量，则面积函数相

当于其累积分布函数,而灰度直方图则相当于其概率密度函数。

对于离散函数,固定 ΔD 为 1,则上述定义为

$$H(D) = A(D) - A(D+1) \tag{2-14}$$

对于数字图像,任一灰度级 D 的面积函数就是大于或等于该灰度值的像素个数。

2. 直方图的性质

直方图具有以下性质:

(1) 直方图是一幅图像中各像素灰度值出现次数或频数的统计结果,它只反映该图像中不同灰度值出现的频率,而不能反映某一灰度值像素所在的位置信息。即直方图仅包含了该图像中某一灰度值的像素出现的概率,而丢失了其所在位置的信息。

(2) 任一幅图像,都能唯一地确定一个与之对应的直方图。但不同的图像,可能有相同的直方图。也就是说,图像与直方图之间是一种多对一的映射关系。

(3) 由于直方图是对具有相同灰度值的像素统计计数得到的,若某一幅图像由若干子图像区域构成,那么各子区域直方图之和就等于原图像的直方图。

2.8.2 直方图的作用

1. 数字化参数

直方图给出了一个简单、直观的可视化指标,可用于判断一幅图像是否合理地利用了全部被允许的灰度级范围。若图像亮度具有超出数字化器所能量化的范围,则这些灰度级将被简单地置为 0 或 255,这时,根据直方图的定义和意义,在其一端或两端将产生尖峰。对直方图的快速检查可以使数字化中产生的问题及早暴露出来,以便及时纠正。

2. 选择边界阈值

假定一幅图像背景是浅色的,其中有一个深色的物体,图 2-17 所示为这类图像的直方图。物体中的深色像素产生了直方图上的左峰,而背景中大量的灰度级产生了直方图上的右峰。物体边界处处于两个峰值之间灰度级的像素数目相对较少,从而产生了两峰之间的谷。选择谷作为灰度阈值将得到合理的物体的边界。

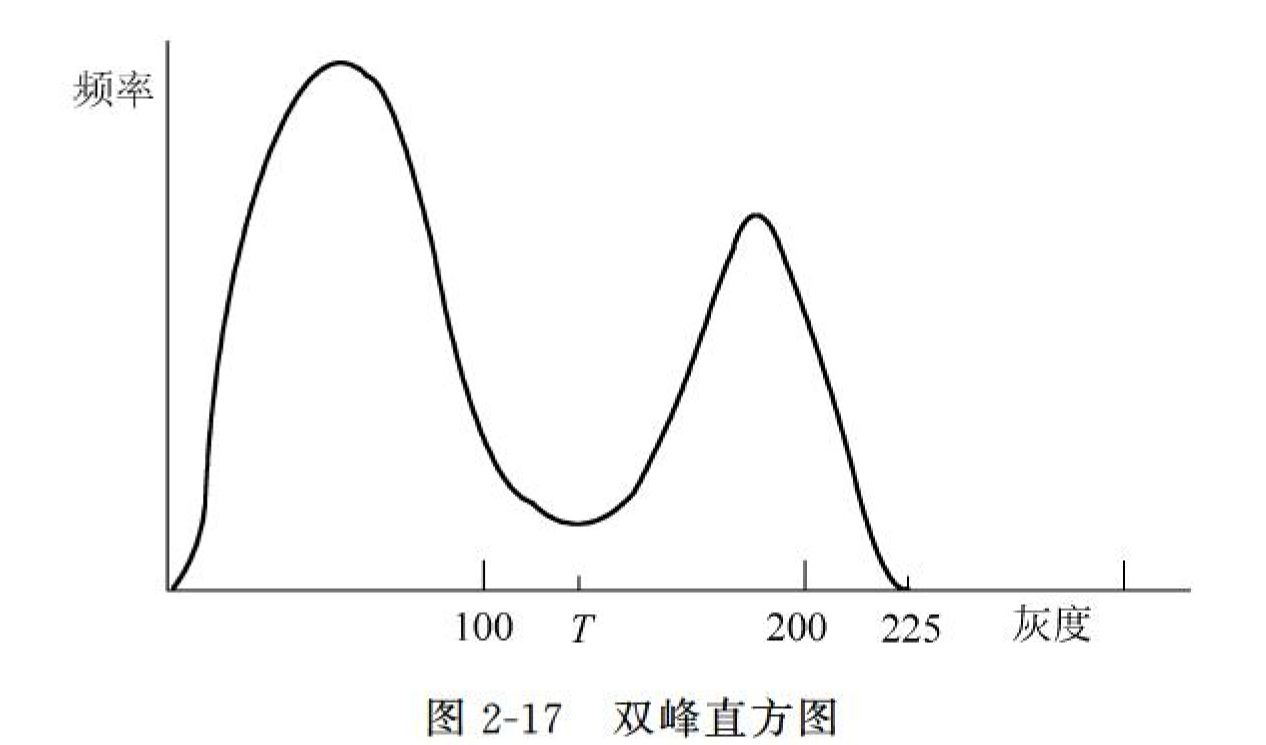


图 2-17 双峰直方图

在某种意义上来说,选择对应于两峰之间的最低点的灰度值作为阈值来确定边界是较合适的。由式(2-13)可知,直方图是面积函数导数的负值。在谷底的附近,直方图的值相对较小,意味着面积函数随阈值灰度级的变化很缓慢。如果选择谷底处的灰度作为阈值,将可以使其对图像内物体的边界的影响达到最小。如果欲测量原图像的面积,选择谷底处作为

阈值可使测量对于阈值灰度变化的敏感度降至最小。

3. 计算综合光密度

某些情况下，综合光密度是图像质量一个很有用的度量，利用直方图可直接计算出综合光密度值。综合光密度用 IOD 来表示，以二维图像为例，设 $D(x,y)$ 表示二维图像的灰度分布，则综合光密度为

$$\mathrm{IOD}=\int_0^a\int_0^b D(x,y)\mathrm{d}x\mathrm{d}y \tag{2-15}$$

其中，a 和 b 是图像区域的边界。对于数字图像，则有

$$\mathrm{IOD}=\sum_{i=1}^{N_L}\sum_{j=1}^{N_S}D(i,j) \tag{2-16}$$

其中，$D(i,j)$ 是像素点 (i,j) 处的灰度值。N_L 和 N_s 分别是图像行和列的数目。令 N_k 代表灰度级为 k 时所对应的像素的个数，则式(2-16)可以化为

$$\mathrm{IOD}=\sum_{k=0}^{255}kN_k$$

由于 $N_k=H(k)$，因此有

$$\mathrm{IOD}=\sum_{k=0}^{255}kH(k) \tag{2-17}$$

即综合光密度是用灰度级加权的直方图之和。

令式(2-16)等于式(2-17)，同时使灰度级的增量趋于 0，则可以得到适用于连续图像的综合光密度公式为

$$\mathrm{IOD}=\int_0^{\infty}DH(D)\mathrm{d}D \tag{2-18}$$

以及

$$\int_0^a\int_0^b D(x,y)\mathrm{d}x\mathrm{d}y=\int_0^{\infty}DH(D)\mathrm{d}D \tag{2-19}$$

若图像中的物体被阈值灰度级为 T 的边界描绘出来，则物体边界内的 IOD 值可以采用如下公式计算，即

$$\mathrm{IOD}(T)=\int_T^{\infty}DH(D)\mathrm{d}D \tag{2-20}$$

内部灰度级的平均值 MGL 等于 IOD 与面积之比值，即

$$\mathrm{MGL}=\frac{\mathrm{IOD}(T)}{A(T)}=\frac{\int_T^{\infty}DH(D)\mathrm{d}D}{\int_T^{\infty}H(D)\mathrm{d}D} \tag{2-21}$$

2.8.3　直方图与图像的关系

对于任一幅图像，直方图是唯一的，若已知一幅图像的函数形式，则可计算其直方图。由于直方图是面积函数关于灰度级的导数的负值。因此，如果能从图像本身的表达式得出面积函数，则有可能获得直方图。如图 2-18 所示，设图像是中心在原点的圆对称高斯脉冲，则图像函数的极坐标表示为

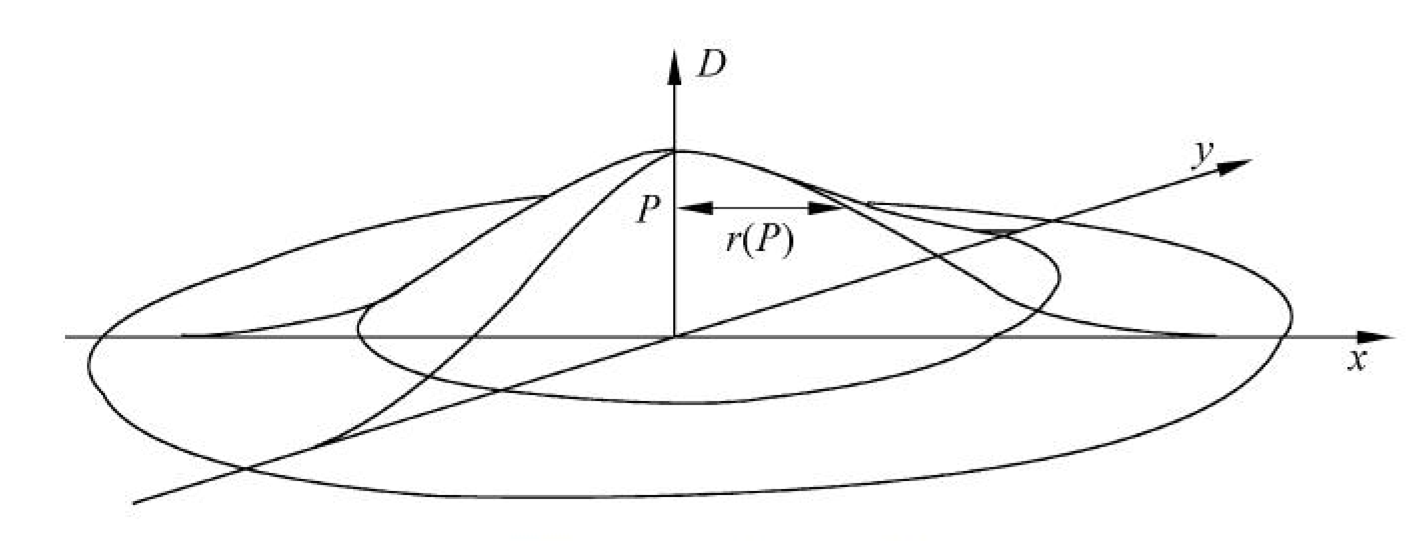


图 2-18 圆形高斯点

$$D(r,\theta)=\mathrm{e}^{-r^2},\quad 0\leqslant\theta\leqslant 2\pi,\quad 0\leqslant r<\infty \tag{2-22}$$

由于灰度级为常数 P 的轮廓线是半径为 $r(P)$ 的圆，根据式(2-22)可得

$$\ln(P)=-r^2$$

因此可得

$$r(P)=\sqrt{-\ln(P)}$$

该轮廓线所包围的面积为

$$A(P)=\pi[r(P)]^2=-\pi\ln(P)$$

对面积函数进行微分，可得到直方图函数，即

$$H(P)=-\frac{\mathrm{d}}{\mathrm{d}P}A(P)=\frac{\pi}{P}$$

曲线如图 2-19 所示。

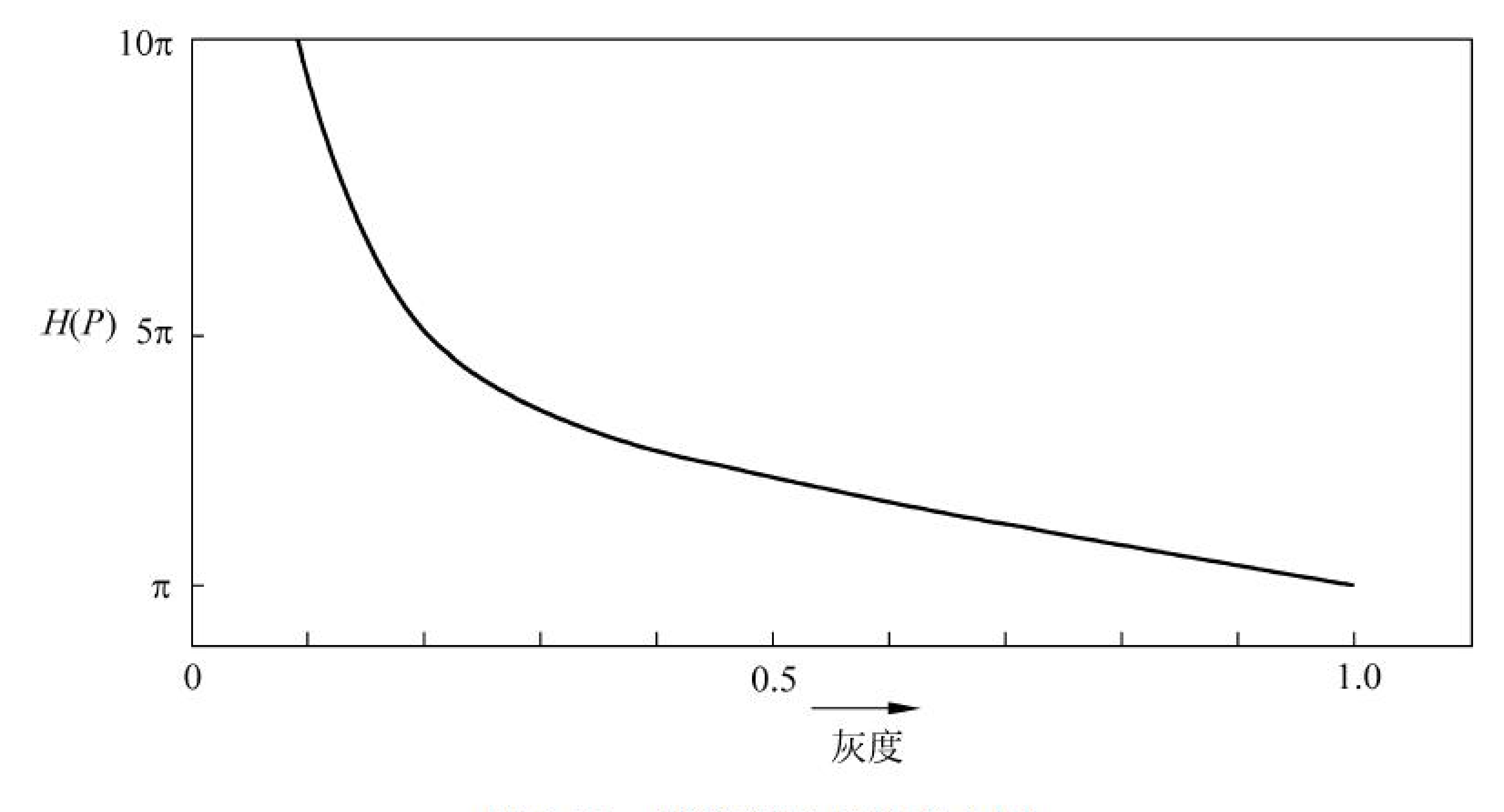


图 2-19 圆形高斯点的直方图

2.9 图像的统计特征

图像反映了自然界中某一物体或对象的电磁波辐射能量分布情况，由于成像系统具有一定的复杂性以及成像过程的随机性，图像信号 $f(x,y)$ 表现出随机变量的特性。因此，图像信息具有随机信号的性质并且具有统计性质，因此统计分析是数字图像处理分析的基本方法之一。

2.9.1 图像的基本统计分析量

设 $f(i,j)$表示大小为 $M\times N$ 的数字图像,则该图像的基本统计量如下。

1. 图像的信息量

一幅图像如果共有 k 种灰度值,并且各灰度值出现的概率分别为 $p_1,p_2,p_3,\cdots,p_k$,根据香农定理,图像的信息量可采用如下公式计算:

$$H=-\sum_{i=1}^{k}p_i\log_2 p_i \tag{2-23}$$

式中,H 称为熵,当图像中各灰度值出现的概率彼此相等时,图像的熵最大。信息量表示一幅图像所含信息的多少,常用于对不同图像处理方法进行比较。例如,对于一幅采用 8bit 表示的数字图像,其信息量为

$$H=-\sum_{i=0}^{255}p_i\log_2 p_i \tag{2-24}$$

2. 图像灰度平均值

灰度均值是指一幅图像中所有像元灰度值的算术平均值。根据算术平均的意义,计算公式为

$$\bar{f}=\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}f(i,j)}{MN} \tag{2-25}$$

图像灰度平均值反映了图像中物体不同部分的平均反射强度。

3. 图像灰度众数

顾名思义,图像灰度众数是指图像中出现次数最多的灰度值。其物理意义是指一幅图像中面积占优的物体的灰度值信息。

4. 图像灰度中值

图像灰度中值是指数字图像全部灰度级中处于中间的值,当灰度级数为偶数时,则取中间的两个灰度值的平均值。例如,若某一图像全部灰度级为 188、176、171、166、160,则灰度中值为 171。

5. 图像灰度方差

灰度方差反映各像元灰度值与图像平均灰度值的离散程度,计算公式为

$$S=\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[f(i,j)-\bar{f}]^2}{MN} \tag{2-26}$$

与熵类似,图像灰度方差同样是衡量图像信息量大小的主要度量指标,是图像统计特性中最重要的统计量之一,方差越大,图像的信息量越大。

6. 图像灰度值域

图像的灰度值域是指图像最大灰度值和最小灰度值之差,计算公式为

$$f_{\text{range}}(i,j)=f_{\max}(i,j)-f_{\min}(i,j) \tag{2-27}$$

2.9.2 多维图像的统计特性

数字图像处理中,一幅 RGB 彩色图像包含了 3 个波段,而一幅遥感图像则可包含多达 7 个波段的灰度图像。对于多波段图像处理,不仅要考虑单个波段图像的统计特性,还应考

虑波段间存在的关联特征。图像波段之间的关联特性不仅是图像分析的重要参数，而且也是图像彩色合成方案的主要依据之一。

1. 协方差

设 $f(i,j)$ 和 $g(i,j)$ 表示大小为 $M\times N$ 的两幅图像，则两者之间的协方差计算公式为

$$S_{gf}^2=S_{fg}^2=\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[f(i,j)-\bar{f}][g(i,j)-\bar{g}] \tag{2-28}$$

式中，$\bar{f}$ 和 $\bar{g}$ 分别表示 $f(i,j)$ 和 $g(i,j)$ 的均值。N 个波段相互间的协方差矩阵用 $\boldsymbol{\Sigma}$ 表示，其定义形式为

$$\boldsymbol{\Sigma}=\begin{bmatrix} S_{11}^2 & S_{12}^2 & \cdots & S_{1N}^2 \\ S_{21}^2 & S_{22}^2 & \cdots & S_{2N}^2 \\ \cdots & \cdots & \ddots & \cdots \\ S_{N1}^2 & S_{N2}^2 & \cdots & S_{NN}^2 \end{bmatrix} \tag{2-29}$$

2. 相关系数

根据概率论与数理统计学知识，数字图像处理技术中的相关系数反映了两个不同波段图像所含信息的重叠程度，它是表示图像不同波段间相关程度的统计量。如果两个波段间的相关系数较大，则表明两个波段具有紧密的相关性，一个波段与其本身的相关系数为 1，表明相关程度达到最大值。当相关系数非常大时，仅选择其中的一个波段就可以表示两个波段的信息。相关系数的计算公式为

$$r_{fg}=\frac{S_{fg}^2}{S_{ff}S_{gg}} \tag{2-30}$$

式中，S_{ff}、S_{gg} 分别表示图像 $f(i,j)$、$g(i,j)$ 的标准差；S_{fg}^2 为图像 $f(i,j)$、$g(i,j)$ 的协方差。N 个波段的相关系数矩阵（简称为相关矩阵）$\boldsymbol{R}$ 定义为

$$\boldsymbol{R}=\begin{bmatrix} 1 & r_{12} & r_{13} & \cdots & r_{1N} \\ r_{21} & 1 & r_{23} & \cdots & r_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & r_{N3} & \cdots & 1 \end{bmatrix} \tag{2-31}$$

习题

1. 数字图像的基本类型有哪几种？各有何特点？
2. 什么是索引图像和 RGB 彩色图像？两者有何区别和联系？
3. 在图像数字化过程中，量化器起什么作用？
4. 图像数字化器的性能指标主要包括哪些？
5. 常用图像数字化器有哪几种？
6. 试简述什么是均匀采样和非均匀采样。
7. 什么是线性量化和非线性量化？在图像数字化过程中，非线性量化具有哪些主要特点？
8. 简要分析图像数字化时如何选择采样和量化参数。
9. 什么是像素的 4 邻域、8 邻域和对角邻域？

10. 常用的数字图像的距离度量有几种？各有何特点？

11. 常见的数字图像格式有哪些？各有何特点？

12. BMP 图像文件的特点有哪些？它由哪儿部分组成？各部分的功能是什么？

13. BMP 文件的文件头包含哪些信息？FILEHEADER 结构占多少字节？这些字节长度是如何分配的？

14. JPEG 图像文件有何特点？它由哪儿部分组成？各部分的功能是什么？

15. GIF 和 PNG 各有何特点？简要描述其优缺点。

16. 什么是半调输出技术？半调输出技术有几种实现类型？

17. 采用半调输出技术时，要实现 16 级灰度效果，需采用什么模板？

18. 试简述半调输出技术的特点，说明采用半调输出对图像的空间分辨率有何影响。

19. 图像质量评价的方法是如何分类的？

20. 图像质量评价的客观方法中常用的有哪些？各有何特点？

21. 什么是图像的灰度直方图？灰度直方图具有哪些性质和作用？

22. 什么是图像的信息量与灰度方差？

第3章 图像变换

CHAPTER 3

人类视觉感受最直接和直观的是空间域和时间域信号，但某些时候，一些问题在空间域和时间域，其特点不明显，不容易观察，而将信号变换到其他域，则特征十分明显，甚至使问题迎刃而解。对于数字图像处理，其变换域分析方法中，频域变换法是应用最为广泛的一类方法。在频域中也有多种变换，如常用的傅里叶变换、DCT变换、小波变换等，每一种变换方法的适应对象和侧重解决的问题各不相同，但无论采用哪种变换，基本目的都相同，即所采用的这种变换一定可以更容易、更方便，或者是更直接、更直观地解决所遇到的图像处理问题。

数字图像处理技术是一门应用性非常强的学科，它既有非常广泛的技术基础，如信息技术、计算机科学、光学技术等学科，也具有严密的数学理论基础。和一些其他应用学科一样，如果没有数学或者说离开了数学，数字图像处理这门学科将难以发展甚至可能不存在。因此，在解决数字图像处理的具体问题时，数学作为图像变换的工具，发挥了重要的作用。

本章将围绕变换法在图像处理中的应用，首先介绍傅里叶变换及二维傅里叶变换的性质和应用，然后介绍在数字图像处理中得到广泛应用的DCT变换、K-L变换、沃尔什变换、哈达玛变换及小波变换等。

3.1 傅里叶变换

傅里叶变换是非常重要的数学分析工具，同时也是一种非常重要的信号处理方法，在图像处理领域，它也是一类应用最为广泛的正交变换，它除了许多在工程上具有重要意义的独特性质之外，还具有快速算法(FFT)。傅里叶变换是线性系统分析的有力工具，在数字图像处理与分析中，图像增强、图像恢复、图像编码压缩、图像分析与描述等每一种处理手段和方法都可以应用图像变换方法。例如，在进行图像低通滤波、高通滤波时，可以借助于傅里叶变换将在空间域中解决的问题转换到频率域中解决。图像处理中的变换方法一般都是保持能量守恒的正交变换，而且在理论上，它的基本运算是严格可逆的。借助于傅里叶变换理论及其物理解释，并结合其他技术学科可以解决或解释大多数图像处理问题。

3.1.1 连续傅里叶变换

1. 一维连续傅里叶变换

若 $f(x)$ 为一维连续实函数，则它的傅里叶变换可定义为

$$F(u)=\int_{-\infty}^{\infty}f(x)\mathrm{e}^{-\mathrm{j}2\pi ux}\mathrm{d}x \tag{3-1}$$

一般情况下，实函数 $f(x)$经过傅里叶变换之后，变换函数 $F(u)$是一个复函数。傅里叶变换是一个线性积分变换，因此应讨论积分变换本身的存在性问题。傅里叶变换在数学上的定义是严密的，它需要满足如下狄利克莱条件：

(1) 具有有限个间断点。

(2) 具有有限个极值点。

(3) 绝对可积。

即只要满足上述条件的函数，其傅里叶变换与逆变换一定是存在的。实际应用中，绝大多数函数都是满足狄利克莱可积条件的。任何图像数字化信号或相关图像信号一般都被截为有限延续且有界的信号（函数），因此，常用的图像信号和函数也都存在傅里叶变换。如果已知 $F(u)$，则其反变换（傅里叶逆变换）为 $f(x)$。傅里叶逆变换定义为

$$f(x)=\int_{-\infty}^{\infty}F(u)\mathrm{e}^{\mathrm{j}2\pi ux}\mathrm{d}u \tag{3-2}$$

除了积分函数和积分变量的区分之外，正变换和反变换在形式上的另一个重要区别是幂次方的符号不同。

函数 $f(t)$和 $F(u)$称为傅里叶变换对。即对于任一函数 $f(x)$，其傅里叶变换 $F(u)$是唯一的；反之，对于任一函数 $F(u)$，其傅里叶逆变换 $f(x)$也是唯一的。

连续函数 $f(x)$的傅里叶变换 $F(u)$是一个复函数，因此 $F(u)$可以表示为

$$F(u)=R(u)+\mathrm{j}I(u)$$

式中，$R(u)$和 $I(u)$分别表示 $F(u)$的实部和虚部，$F(u)$也可以表示为指数形式，即

$$F(u)=|F(u)|\mathrm{e}^{\mathrm{j}\phi(u)}$$

式中

$$|F(u)|=[R^2(u)+I^2(u)]^{\frac{1}{2}} \tag{3-3}$$

$$\phi(u)=\arg\tan\left(\frac{I(u)}{R(u)}\right) \tag{3-4}$$

式中，$|F(u)|$称为 $F(u)$的模，也称为函数 $f(x)$的傅里叶谱；$\phi(u)$为 $F(u)$的相角，称为相位谱。

令

$$E(u)=|F(u)|^2 \tag{3-5}$$

则 $E(u)$称为函数 $f(x)$的能量谱或功率谱。

2. 二维连续傅里叶变换

若 $f(x,y)$为二维连续函数，并满足可积条件，则它的傅里叶变换可定义为

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y \tag{3-6}$$

式中，u 是对应于 x 轴的空间频率变量；v 是对应于 y 轴的空间频率变量。

一般情况下，$F(u,v)$是关于实变量 u、v 的复值函数。由于一幅图像可用二维函数 $f(x,y)$表示，所以 $F(u,v)$也就是二维图像 $f(x,y)$的傅里叶变换或傅里叶频谱。

如果已知 $F(u,v)$，且 $F(u,v)$满足可积条件，则其傅里叶逆变换定义为

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(u,v)\mathrm{e}^{\mathrm{j}2\pi(ux+vy)}\mathrm{d}u\mathrm{d}v \tag{3-7}$$

这时 $F(u,v)$和 $f(x,y)$称为傅里叶变换对。类似于一维傅里叶变换,二维傅里叶频谱也可以表示为

$$F(u,v)=R(u,v)+\mathrm{j}I(u,v) \tag{3-8}$$

式中,$R(u,v)$和 $I(u,v)$分别表示 $F(u,v)$的实部和虚部。$F(u,v)$也可以表示为指数形式,即

$$F(u,v)=|F(u,v)|\mathrm{e}^{\mathrm{j}\phi(u,v)}$$

式中

$$|F(u,v)|=[R^2(u,v)+I^2(u,v)]^{\frac{1}{2}} \tag{3-9}$$

$$\phi(u)=\operatorname{argtan}\left(\frac{I(u,v)}{R(u,v)}\right) \tag{3-10}$$

式中,$|F(u,v)|$称为 $F(u,v)$的模,也称为函数 $f(x,y)$的幅值谱;$\phi(u,v)$为 $F(u,v)$的相角,称为相位谱。

令

$$E(u,v)=|F(u,v)|^2 \tag{3-11}$$

则 $E(u,v)$称为函数 $f(x,y)$的能量谱或功率谱。

一维连续函数的傅里叶变换的许多结论都可以很容易地根据定义推广到二维傅里叶变换。

例如,对于二维函数

$$f(x,y)=\begin{cases}A, & |x|\leqslant\dfrac{T}{2}\ |y|\leqslant\dfrac{T}{2}\\[2ex] 0, & |x|>\dfrac{T}{2}\ |y|>\dfrac{T}{2}\end{cases} \tag{3-12}$$

其几何图形如图 3-1 所示。

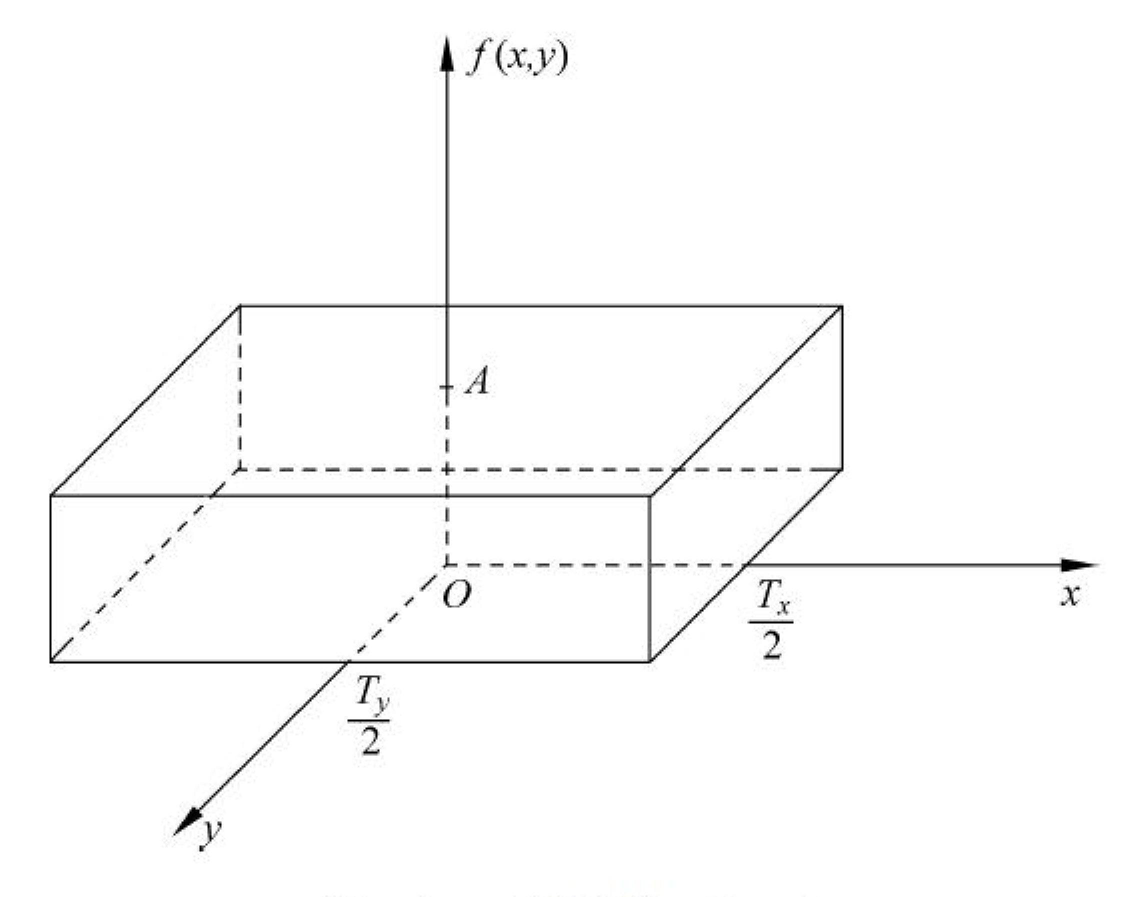


图 3-1 二维函数 $f(x,y)$

根据傅里叶变换的定义,二维图像 $f(x,y)$傅里叶变换为

$$\begin{aligned}F(u,v)&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y\\&=\int_{-\frac{T_x}{2}}^{\frac{T_x}{2}}\int_{-\frac{T_y}{2}}^{\frac{T_y}{2}}A\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y\\&=A\int_{-\frac{T_x}{2}}^{\frac{T_x}{2}}\mathrm{e}^{-\mathrm{j}2\pi ux}\mathrm{d}x\int_{-\frac{T_y}{2}}^{\frac{T_y}{2}}\mathrm{e}^{-\mathrm{j}2\pi vy}\mathrm{d}y\\&=AT_xT_y\frac{\sin(\pi uT_x)}{\pi uT_x}\mathrm{e}^{-\mathrm{j}\pi uT_x}\frac{\sin(\pi vT_y)}{\pi vT_y}\mathrm{e}^{-\mathrm{j}\pi vT_y}\end{aligned}\tag{3-13}$$

因此,可得函数 $f(x,y)$的傅里叶频谱为

$$|F(u,v)|=AT_xT_y\left|\frac{\sin(\pi uT_x)}{\pi uT_x}\right|\left|\frac{\sin(\pi vT_y)}{\pi vT_y}\right|\tag{3-14}$$

若 T_x 与 T_y 相等,则二维函数 $f(x,y)$的频谱 $F(u,v)$的几何图形可以通过将一维矩形信号的频谱旋转一周实现,或者说,一维矩形信号的频谱是二维频谱 $F(u,v)$的径向剖面图形。

3.1.2　离散傅里叶变换

离散傅里叶变换(DFT)是指对离散序列进行傅里叶变换。由于计算机能处理的数据为数字量或者说离散量,而且二维数字图像已经将连续图像离散化为像素点,因此,连续函数的傅里叶变换在计算机上无法直接使用。因此,为了能在计算机上实现数字图像的傅里叶变换,必须将连续函数离散化。连续函数的傅里叶变换是波形分析的有力工具,离散函数的傅里叶变换将数学与计算机技术紧密联系在一起,为傅里叶变换这一强有力的数学工具开辟了一条宽阔的实用之道,使傅里叶变换这一数学工具将发挥更大的作用。

1. 一维离散傅里叶变换

前面介绍了一维连续函数 $f(x)$的傅里叶变换,$f(x)$是在$-\infty\sim+\infty$无限区间上连续,函数值 f 是连续的,计算机无法直接对连续函数进行运算。因此,必须对连续函数 $f(x)$进行离散化处理。

若以 Δx 为采样间隔,从$-\infty\sim+\infty$对 $f(x)$进行等间隔采样,则可将连续函数离散化。一般情况下不会对这无穷多个采样值进行同样的关注,若以某个起点 x_0 开始的采样值是所关注的值,则称该起点 x_0 的采样值为离散采样序列的第 1 个样本值,其余采样点以此类推,$(x_0+\Delta x)$点的采样值为第 2 个采样值,$(x_0+2\Delta x)$点的采样值为第 3 个采样值……$[x_0+(N-1)\Delta x]$点处的采样值为第 N 个采样值。这样就得到了具有 N 个采样值的离散序列。将 N 个采样值排列如下:

$$f(x_0),f(x_0+\Delta x),f(x_0+2\Delta x),\cdots,f[x_0+(N-1)\Delta x]$$

上述序列可以表示为

$$f(x_0+n\Delta x),\quad n=0,1,2,3,\cdots,N-1$$

由于 x_0 是一个确定的起点时刻,Δx 是采样间隔,这两个量都是常量,上述序列的表达式中只有 n 是变量,因此,离散采样序列可以直接表示为 $f(n)$,即

$$f(n)=f(x_0+n\Delta x),\quad n=0,1,2,3,\cdots,N-1\tag{3-15}$$

为了和数字图像的其他表示方法一致,可以将 x 代替 n,即序列可以表示为

$$f(x)=f(x_0+x\Delta x),\quad x=0,1,2,3,\cdots,N-1\tag{3-16}$$

由此可得一维离散序列 $f(x)(x=0,1,2,3,\cdots,N-1)$的傅里叶变换定义为

$$F(u)=\sum_{x=0}^{N-1}f(x)\mathrm{e}^{-\mathrm{j}\frac{2\pi ux}{N}},\quad u=0,1,2,3,\cdots,N-1 \tag{3-17}$$

式中

$$F(u)=F(u_0+u\Delta u),\quad u=0,1,2,3,\cdots,N-1$$

若已知频率序列 $F(u)(u=0,1,2,3,\cdots,N-1)$，则离散序列 $F(u)$的傅里叶逆变换定义为

$$f(x)=\frac{1}{N}\sum_{u=0}^{N-1}F(u)\mathrm{e}^{\mathrm{j}\frac{2\pi ux}{N}},\quad x=0,1,2,3,\cdots,N-1 \tag{3-18}$$

式中，$f(x)$和 $F(u)$称为傅里叶变换对；Δx 和 Δu 分别为空间域采样间隔和频率域采样间隔，两者之间满足

$$\Delta x=\frac{1}{N\Delta u} \tag{3-19}$$

令

$$W=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}} \tag{3-20}$$

离散傅里叶变换可以写为如下形式：

正变换：

$$F(u)=\sum_{x=0}^{N-1}f(x)W_N^{ux},\quad u=0,1,2,3,\cdots,N-1 \tag{3-21}$$

逆变换：

$$f(x)=\sum_{u=0}^{N-1}F(u)W_N^{-ux},\quad x=0,1,2,3,\cdots,N-1 \tag{3-22}$$

根据欧拉公式，傅里叶变换可以写为

$$F(u)=\sum_{x=0}^{N-1}f(x)\left[\cos\frac{2\pi ux}{N}-\mathrm{j}\sin\frac{2\pi ux}{N}\right],\quad u=0,1,2,3,\cdots,N-1$$

由此可知，离散序列 $f(x)$的傅里叶变换 $F(u)$依然是离散序列，而且通常情况下是一个复数序列，与连续傅里叶变换类似，$F(u)$可以表示为

$$F(u)=R(u)+\mathrm{j}I(u)$$

式中，序列 $R(u)$和 $I(u)$分别表示离散序列 $F(u)$的实序列和虚序列，序列 $F(u)$还可以表示为指数形式，即

$$F(u)=|F(u)|\mathrm{e}^{\mathrm{j}\phi(u)} \tag{3-23}$$

式中

$$|F(u)|=[R^2(u)+I^2(u)]^{\frac{1}{2}} \tag{3-24}$$

$$\phi(u)=\mathrm{argtan}\left(\frac{I(u)}{R(u)}\right) \tag{3-25}$$

式中，$|F(u)|$称为 $F(u)$的模，又称为序列 $f(x)$的频谱或傅里叶幅度谱；$\phi(u)$称为 $F(u)$的相角，或称为序列 $f(x)$的相位谱。

令

$$E(u)=|F(u)|^2 \tag{3-26}$$

则频谱的平方 $E(u)$称为序列 $f(x)$的能量谱或功率谱。

2. 二维离散傅里叶变换

根据一维离散傅里叶变换的定义和二维连续傅里叶变换理论，对于一个具有 $M\times N$ 个样本值的二维离散序列 $f(x,y)(x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1)$，其傅里叶变换为

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)},\quad u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-27}$$

式中

$$F(u,v)=F(u_0+u\Delta u,v_0+v\Delta v),\quad u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1$$

若已知频率二维序列 $F(u,v)(u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1)$，则二维离散序列 $F(u,v)$ 的傅里叶逆变换定义为

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)},\quad x=0,1,2,3,\cdots,M-1\quad y=0,1,2,3,\cdots,N-1 \tag{3-28}$$

式中，u 是对应于 x 轴的空间频率分量；v 是对应于 y 轴的空间频率分量。

$f(x,y)$ 和 $F(u,v)$ 称为傅里叶变换对，Δx 和 Δu 分别为空间域采样间隔和频率域采样间隔，两者之间满足

$$\begin{cases}\Delta x=\dfrac{1}{M\Delta u}\\[2ex]\Delta y=\dfrac{1}{N\Delta v}\end{cases} \tag{3-29}$$

根据欧拉公式，二维离散序列 $f(x,y)$ 的傅里叶变换 $F(u,v)$ 依然是二维离散复数序列，$F(u,v)$ 可以表示为

$$F(u,v)=R(u,v)+\mathrm{j}I(u,v)$$

式中，序列 $R(u,v)$ 和 $I(u,v)$ 分别表示离散序列 $F(u,v)$ 的实序列和虚序列。则同样可得二维序列 $f(x,y)$ 的频谱(傅里叶幅度谱)、相位谱和能量谱(功率谱)分别为

$$|F(u,v)|=[R^2(u,v)+I^2(u,v)]^{\frac{1}{2}} \tag{3-30}$$

$$\phi(u,v)=\operatorname{argtan}\left(\frac{I(u,v)}{R(u,v)}\right) \tag{3-31}$$

$$E(u,v)=|F(u,v)|^2 \tag{3-32}$$

3.1.3 二维 DFT 的性质

根据傅里叶变换的定义，二维傅里叶变换具有与一维傅里叶变换相似的特性。数字图像处理中需要应用很多二维离散傅里叶变换的性质，充分理解和掌握这些性质是非常必要的。二维离散傅里叶变换的主要特点如下。

1. 线性特性

如果二维离散函数 $f_1(x,y)$ 和 $f_2(x,y)$ 的傅里叶变换分别为 $F_1(u,v)$ 和 $F_2(u,v)$，则存在以下线性性质：

$$\begin{aligned}\mathrm{DFT}[k_1f_1(x,y)+k_2f_2(x,y)]&=\mathrm{DFT}[k_1f_1(x,y)]+\mathrm{DFT}[k_2f_2(x,y)]\\&=k_1F_1(u,v)+k_2F_2(u,v)\end{aligned}$$

$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1$$
$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \qquad (3\text{-}33)$$

应用线性性质时应注意，$k_1F_1(u,v)+k_2F_2(u,v)$不能超过图像显示器所允许的最大值，否则可能造成图像信息损失。

2. 比例性质

如果二维离散函数 $f(x,y)$的傅里叶变换为 $F(u,v)$，则存在以下比例性质：

$$\mathrm{DFT}[f(ax,by)]=\frac{1}{ab}F\left(\frac{u}{a},\frac{v}{b}\right),\quad ab\neq 0$$
$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1$$
$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \qquad (3\text{-}34)$$

式中，若取

$$\begin{cases}a=-1\\ b=-1\end{cases}$$

则比例特性表现为

$$\mathrm{DFT}[f(-x,-y)]=F(-u,-v) \qquad (3\text{-}35)$$

3. 平移性质

如果二维离散函数 $f(x,y)$的傅里叶变换为 $F(u,v)$，则存在以下平移性质：

$$\mathrm{DFT}\left[f(x,y)\mathrm{e}^{\mathrm{j}2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)}\right]=F(u-u_0,v-v_0)$$
$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1$$
$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \qquad (3\text{-}36)$$

对于逆变换，也存在同样的性质：

$$\mathrm{DFT}[f(x-x_0,y-y_0)=F(u,v)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{ux_0}{M}+\frac{vy_0}{N}\right)} \qquad (3\text{-}37)$$

二维傅里叶变换的移位特性表明，当用 $\mathrm{e}^{\mathrm{j}2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)}$乘以 $f(x,y)$，然后再进行乘积的离散傅里叶变换时，可以使空间频率域 u-v 平面坐标系的原点从(0,0)平移到(u_0,v_0)的位置。同样，对于傅里叶逆变换，当用 $\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{ux_0}{M}+\frac{vy_0}{N}\right)}$乘以 $F(u,v)$，并求此乘积的离散傅里叶反变换，可以使空间域 x-y 平面坐标系的原点从(0,0)平移到(x_0,y_0)的位置。逆变换的移位特性还表明，图像 $f(x,y)$平移之后为 $f(x-x_0,y-y_0)$，但平移之后的傅里叶幅度谱没有发生任何变化，而仅仅是相位谱产生了一定的相移特性。

二维离散傅里叶变换的移位特性在数字图像处理中具有重要的应用价值。例如，为方便地观察数字图像的傅里叶变换结果，经常需要将空间频率平面坐标系的原点(0,0)移到$(M/2,N/2)$的位置，此时，可以取值为

$$\begin{cases}u_0=\dfrac{M}{2}\\[2mm] v_0=\dfrac{N}{2}\end{cases}$$

根据移位特性，则对应于空间域乘以因子

$$\mathrm{e}^{\mathrm{j}2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)}=\mathrm{e}^{\mathrm{j}\pi(x+y)}=(-1)^{x+y}$$

这时移位特性表现为

$$\mathrm{DFT}\left[f(x,y)\mathrm{e}^{\mathrm{j}2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)}\right]=\mathrm{DFT}\left[(-1)^{x+y}f(x,y)\right]=F(u-u_0,v-v_0)$$

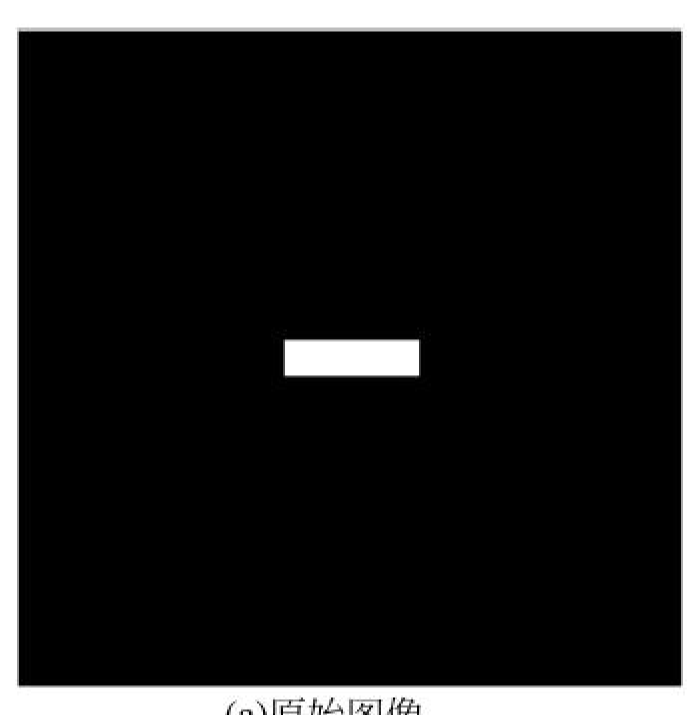
$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1$$

$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-38}$$

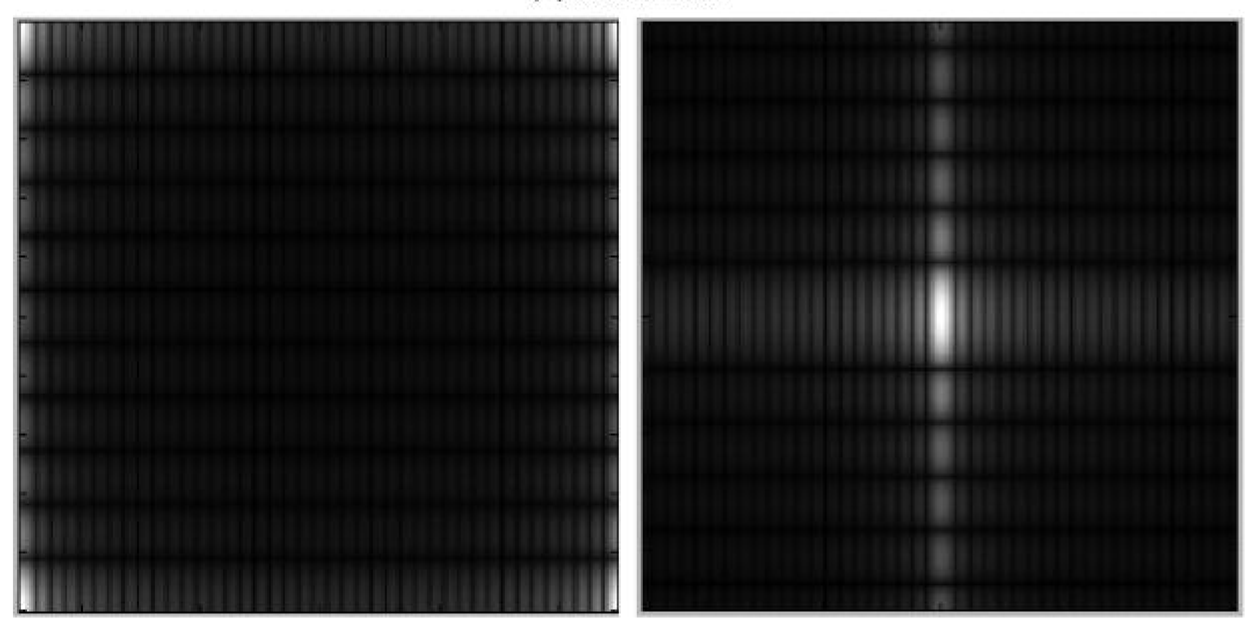
式(3-38)表明，将 $f(x,y)$ 乘以一个简单的因子 $(-1)^{x+y}$，然后对乘积进行傅里叶变换，就可以将空间域频率平面坐标系的原点平移到空间频率域的 $M\times N$ 方阵中心。

例 3-1：傅里叶变换的频谱分布特性

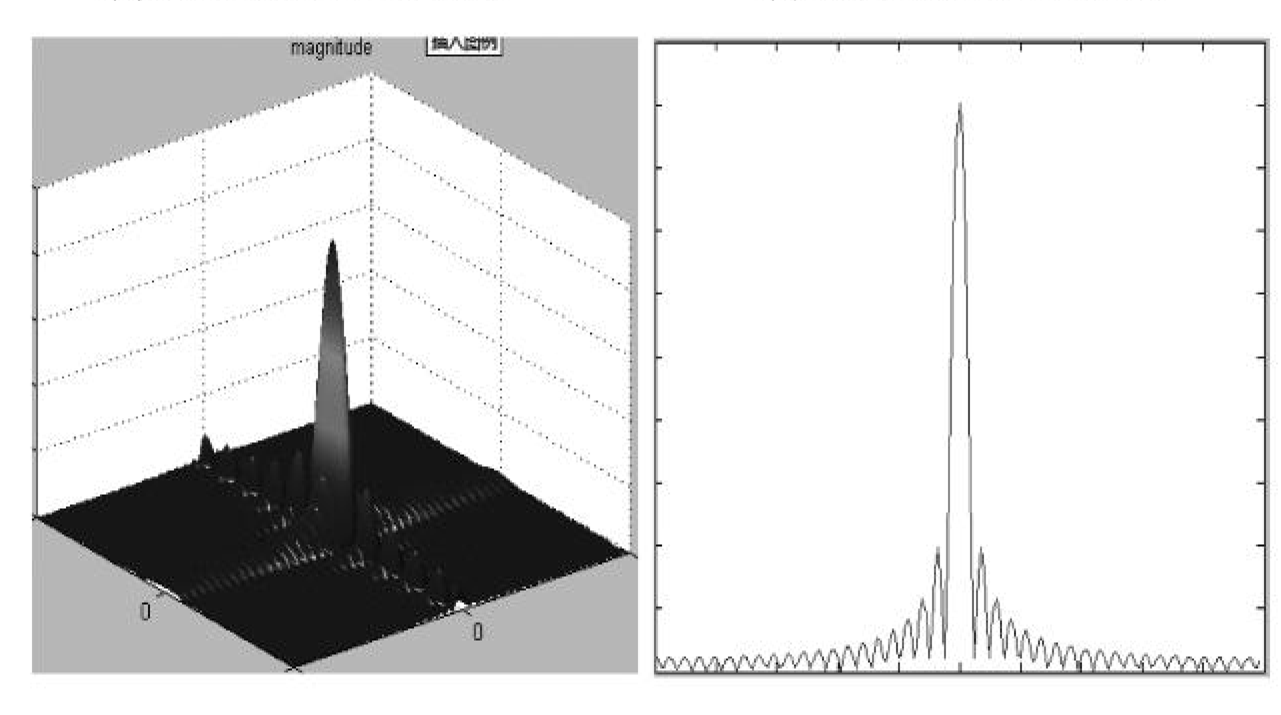
对如图 3-2(a)所示图像进行傅里叶变换，图 3-2(b)为未进行移位的傅里叶变换幅值谱，幅值谱图的四个角为低频点，图 3-2(c)为先进行移位再进行傅里叶变换的幅值谱，频谱的中心位于幅值谱图的中心点，图 3-2(d)为图 3-2(c)幅值谱的三维显示，图 3-2(e)为三维幅值谱沿 u 频率轴剖面。



(a)原始图像



(b) 未移位傅里叶幅值谱　　(c) 移位后傅里叶幅值谱



(d) 移位后幅值谱三维图　　(e) 三维幅值谱沿 u 频率轴剖面

图 3-2　傅里叶变换及移位特性

4. 可分离性

如果 $M\times N$ 二维离散函数 $f(x,y)$ 的傅里叶变换为 $F(u,v)$，则存在以下可分离性质：

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$=\sum_{x=0}^{M-1}\left\{\left[\sum_{y=0}^{N-1}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi\frac{vy}{N}}\right]\mathrm{e}^{-\mathrm{j}2\pi\frac{ux}{M}}\right\}$$

$$u=0,1,2,3,\cdots,M-1,\quad v=0,1,2,3,\cdots,N-1 \tag{3-39}$$

二维傅里叶变换的可分离特性表明，一个二维傅里叶变换可通过二次一维傅里叶变换来完成，即第一次先对 y 进行一维傅里叶变换

$$F(x,v)=\sum_{y=0}^{N-1}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi\frac{vy}{N}}$$

$$x=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-40}$$

然后，再在此基础上对 x 进行一维傅里叶变换

$$F(u,v)=\sum_{x=0}^{M-1}f(x,v)\mathrm{e}^{-\mathrm{j}2\pi\frac{ux}{M}}$$

$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-41}$$

上述过程也可以先对 y 进行傅里叶变换，然后再对 x 进行变换。变量分离步骤如图 3-3 所示。

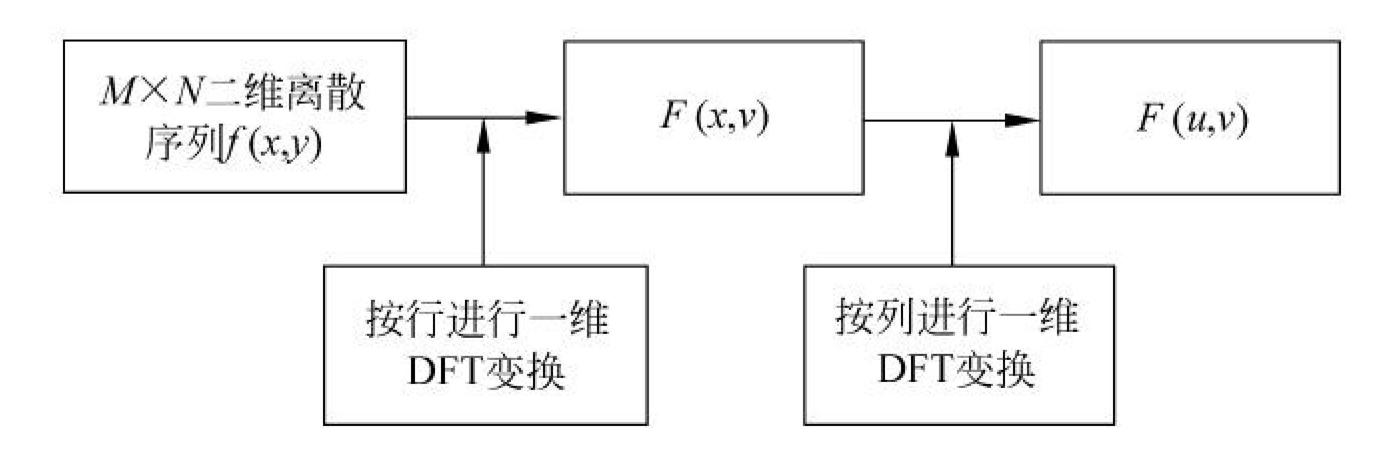


图 3-3 二维离散傅里叶变换的分离特性

根据分离性质可以表示为

$$F(u,v)=\mathrm{DFT}_y\{\mathrm{DFT}_x[f(x,y)]\}=\mathrm{DFT}_x\{\mathrm{DFT}_y[f(x,y)]\} \tag{3-42}$$

若已知频率二维序列 $F(u,v)$，则二维可分离性对傅里叶逆变换同样适应。

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$=\sum_{u=0}^{M-1}\left\{\left[\sum_{v=0}^{N-1}F(u,v)\mathrm{e}^{-\mathrm{j}2\pi\frac{vy}{N}}\right]\mathrm{e}^{-\mathrm{j}2\pi\frac{ux}{M}}\right\}$$

$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1 \tag{3-43}$$

逆变换的分离性也同样可以分解为两次一维傅里叶变换：

$$f(x,y)=\mathrm{DFT}_u^{-1}\{\mathrm{DFT}_v^{-1}[F(u,v)]=\mathrm{DFT}_v^{-1}\{\mathrm{DFT}_u^{-1}[F(u,v)]$$

$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1 \tag{3-44}$$

5. 周期性

如果二维离散函数 $f(x,y)$ 的傅里叶变换为 $F(u,v)$，则傅里叶变换及其逆变换存在如

下周期特性：

$$F(u,v)=F(u+k_1M,v+k_2N)$$
$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-45}$$
$$f(x,y)=F(x+k_1M,y+k_2N)$$
$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1 \tag{3-46}$$

式中，k_1、k_2 均为正整数。

6. 共轭对称性

如果二维离散函数 $f(x,y)$的傅里叶变换为 $F(u,v)$，则傅里叶变换存在共轭对称性：

$$F(u,v)=F^*(-u,-v)$$
$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-47}$$

且

$$|F(u,v)|=|F^*(-u,-v)|$$
$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-48}$$

7. 旋转不变性

如果二维离散函数 $f(x,y)$的傅里叶变换为 $F(u,v)$，则二维傅里叶变换对之间存在旋转不变性。考虑到极坐标表示二维图形的旋转特性的方便性，为此，将空间域和空间频率域都改为用极坐标表示。在空间域直角坐标与极坐标的变换关系为

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases} \tag{3-49}$$

这样，图像 $f(x,y)$可以表示为 $f(r,\theta)$。同样，空间频率域的 $F(u,v)$采用极坐标可以表示为 $F(\rho,\phi)$。二维离散傅里叶存在如下旋转特性：

$$\mathrm{DFT}[f(r,\theta+\theta_0)]=F(\rho,\phi+\theta_0) \tag{3-50}$$

即如果 $f(x,y)$旋转一个角度 θ_0，则对应的傅里叶变换 $F(u,v)$也同样旋转相同的角度 θ_0。反之亦然。

$$\mathrm{DFT}[F(\rho,\phi+\theta_0)]=f(r,\theta+\theta_0) \tag{3-51}$$

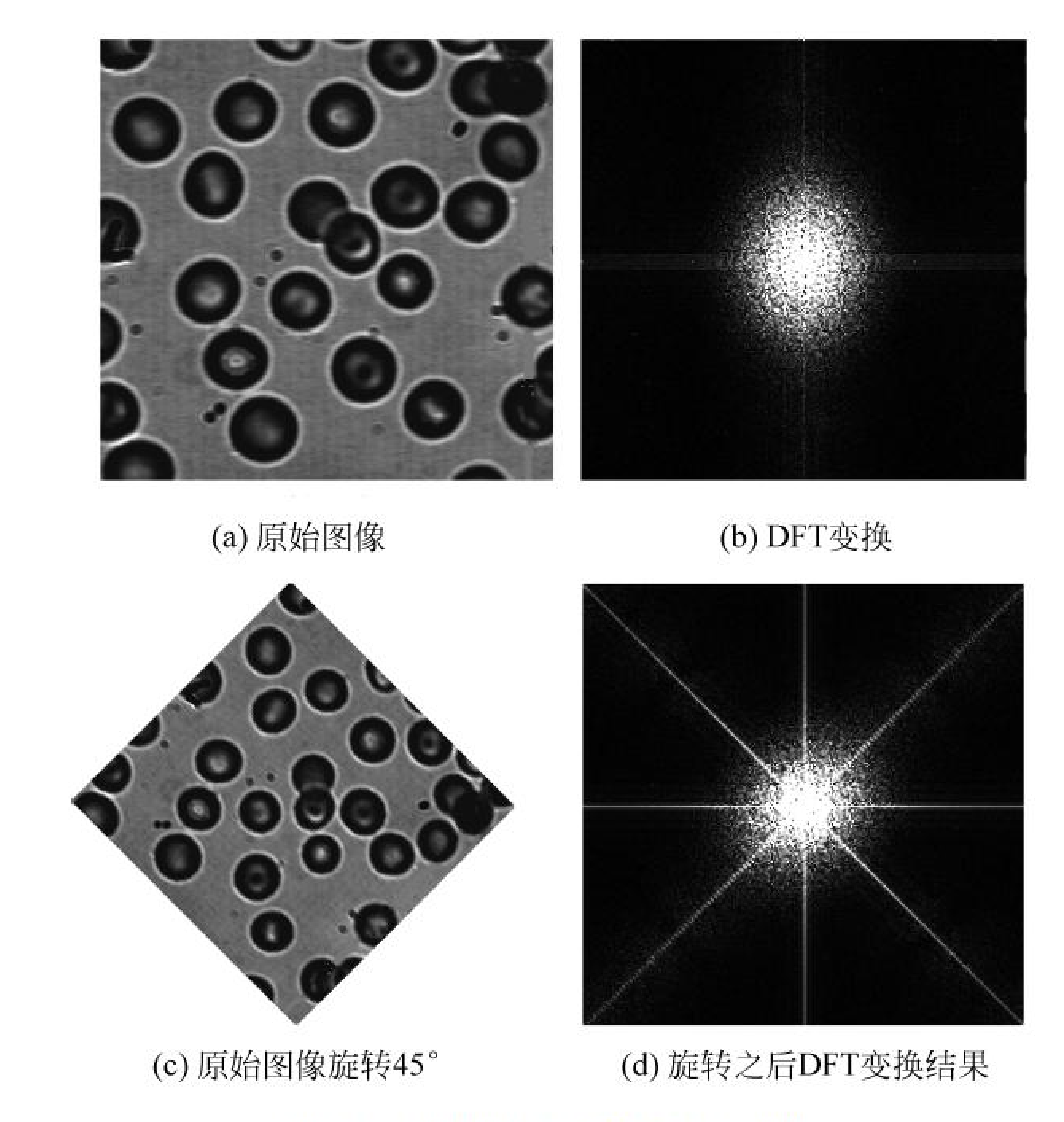
例 3-2：傅里叶变换的旋转不变性

如图 3-4 所示是傅里叶变换旋转不变性的示例，图 3-4(a)为原始图像，图 3-4(b)是对原始图像进行离散傅里叶变换的结果，图 3-4(c)为将原始图像旋转 45°角，图 3-4(d)为旋转之后的傅里叶变换，可以看出图 3-4(b)与图 3-4(d)之间同样是 45°的旋转关系。

8. 微分性质

如果二维离散函数 $f(x,y)$的傅里叶变换为 $F(u,v)$，则傅里叶变换具有如下微分性质。

$$\mathrm{DFT}\left[\frac{\partial^n f(x,y)}{\partial x^n}\right]=(\mathrm{j}2\pi u)^n F(u,v)$$
$$u=0,1,2,3,\cdots,M-1;\quad v=0,1,2,3,\cdots,N-1 \tag{3-52}$$
$$\mathrm{DFT}\left[\frac{\partial^n f(x,y)}{\partial y^n}\right]=(\mathrm{j}2\pi v)^n F(u,v)$$
$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-53}$$



(a) 原始图像　(b) DFT变换

(c) 原始图像旋转45°　(d) 旋转之后DFT变换结果

图 3-4　傅里叶变换的旋转不变性

9. 平均值性质

如果 $M\times N$ 二维离散函数 $f(x,y)$ 的傅里叶变换为 $F(u,v)$，函数的平均值定义如下：

$$\bar{f}(x,y)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y) \tag{3-54}$$

则二维离散傅里叶变换具有如下性质：

$$F(0,0)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)=MN\,\bar{f}(x,y) \tag{3-55}$$

即

$$\bar{f}(x,y)=\frac{1}{MN}F(0,0)$$

也就是说，二维离散函数的平均值等于其傅里叶变换在频率原点处值的 $1/MN$。

10. 卷积定理

如果二维离散函数 $f(x,y)$ 和 $h(x,y)$ 的傅里叶变换分别为 $F(u,v)$ 和 $H(u,v)$，则 $f(x,y)$ 和 $h(x,y)$ 之间的卷积可以通过其傅里叶变换 $F(u,v)$ 和 $H(u,v)$ 进行计算。

$$\mathrm{DFT}[f(x,y)*h(x,y)]=F(u,v)H(u,v) \tag{3-56}$$

反过来，也存在如下关系：

$$\mathrm{DFT}^{-1}[F(u,v)*H(u,v)]=f(x,y)h(x,y) \tag{3-57}$$

11. 相关定理

如果二维离散函数 $f(x,y)$ 的傅里叶变换为 $F(u,v)$，则傅里叶变换对自相关和互相关运算分别具有如下特性：

1）互相关

$$f(x,y) \odot h(x,y) \Leftrightarrow F(u,v) H^{*}(u,v) \tag{3-58}$$

$$F(u,v) \odot H(u,v) \Leftrightarrow f(x,y) h^{*}(x,y) \tag{3-59}$$

2）自相关

$$f(x,y) \odot f(x,y) \Leftrightarrow F(u,v) F^{*}(u,v) = |F(u,v)|^{2} \tag{3-60}$$

$$|f(x,y)|^{2} \Leftrightarrow F(u,v) \odot F(u,v) \tag{3-61}$$

12．帕萨瓦（Parseval）定理

如果 $M \times N$ 二维离散函数 $f(x,y)$ 的傅里叶变换为 $F(u,v)$，则傅里叶变换具有如下性质：

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^{2} \mathrm{d}x\mathrm{d}y = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |F(u,v)|^{2} \mathrm{d}u\mathrm{d}v \tag{3-62}$$

帕萨瓦定理表明，进行二维傅里叶变换后函数的能量没有改变。

3.1.4 图像傅里叶变换综合实例

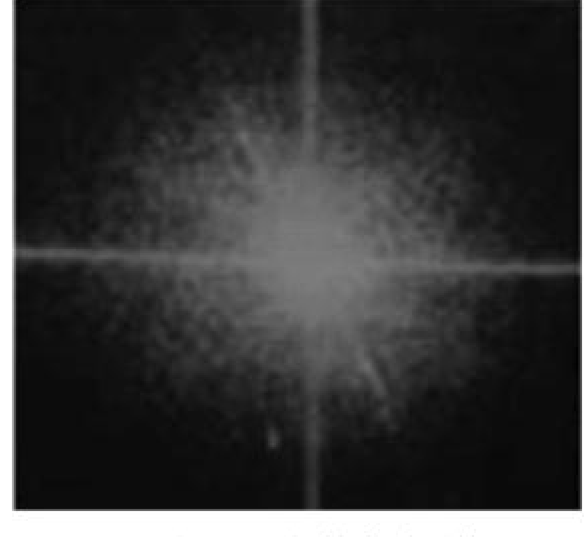
相对于一维信号的傅里叶变换，二维傅里叶变换有一些新特性。为了全面理解和掌握傅里叶变换，现给出一例傅里叶变换的综合实例。

例 3-3：傅里叶变换的综合实例

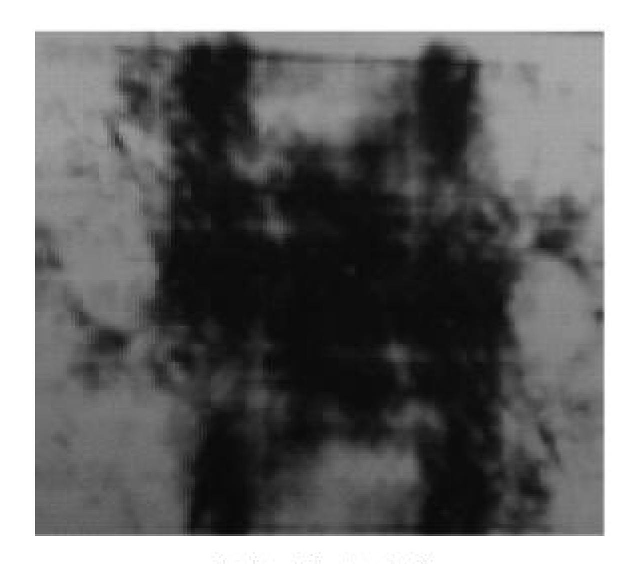
如图 3-5(a)所示为原始图像，图 3-5(b)所示为傅里叶变换的幅值谱，假定相位为常数 0，由幅值谱进行傅里叶逆变换的重建图像如图 3-5(c)所示；图 3-5(d)是傅里叶变换的相位谱，假定幅值为常数，由相位谱逆变换的重建图像如图 3-5(e)所示；由幅值谱和相位谱逆变换的重建图像如图 3-5(f)所示。



(a) 原始图像

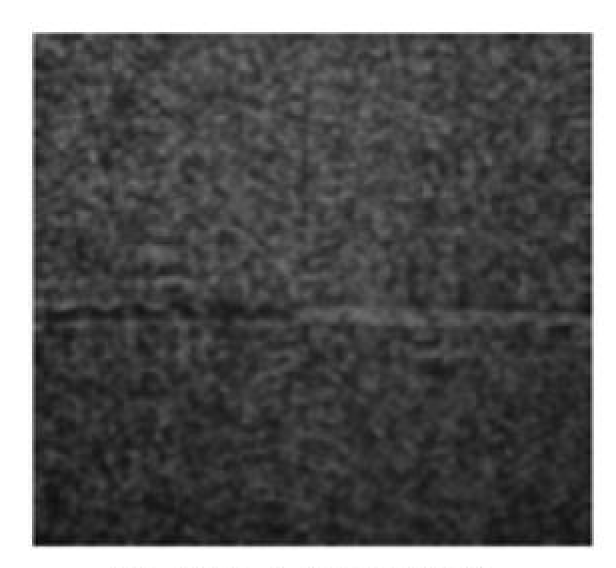


(b) 傅里叶变换幅值谱



(c) 幅值谱逆变换

图 3-5 图像傅里叶变换实例




(d) 傅里叶变换相位谱



(e) 相位谱逆变换



(f) 幅值谱和相位谱重建图像

图 3-5(续)

本实验结果表明，基于相位谱和幅值谱逆变换的图像 3-5(f)和原始图像 3-5(a)误差不大，假定相位为常数 0，由幅值谱进行傅里叶逆变换的重建图像 3-5(c)和原始图像完全不一样，而假定幅值为常数，由相位谱逆变换的重建图像 3-5(d)，虽然与原始图像在灰度值方面有较大的误差，但图像的基本轮廓和边缘却相似。

3.2 离散余弦变换

DFT 变换是频谱分析的有力工具，但 DFT 变换是基于复数域的运算，因而给实际运算带来了不便。因此，工程上特别需要各种在实数域内的变换，DCT 变换就是其中一种。DCT 变换除了具有一般正交变换的性质之外，还具有许多突出的优点。DCT 变换阵的基向量很近似于 Toeplitz 矩阵的特征向量，很好地体现了人类语音信号及图像信号的相关特性。因此，许多学者认为，在语音和图像信号处理方面，DCT 变换被认为是准最佳变换。而且，DCT 变换还可以通过实偶函数的傅里叶变换建立与 FFT 变换之间的关系。

DCT 变换由 Ahmed 和 Rao 于 1974 年首先提出，随后就得到了广泛的应用，在许多领域，DCT 变换被认为是一种准最佳变换。在近年颁布的一系列视频压缩编码的国际标准建议中，都将 DCT 变换作为其中的一个基本处理模块。DCT 变换除上述优点之外，还具有许多特点，如 DCT 为实数变换、变换矩阵确定(与变换对象无关)、具有多种快速算法，二维 DCT 还是一种可分离的变换等。

3.2.1　一维 DCT 变换

一维 DCT 的变换核定义为

$$g(x,u)=C(u)\sqrt{\frac{2}{N}}\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$
$$x,u=0,1,2,3,\cdots,N-1 \tag{3-63}$$

式中

$$C(u)=\begin{cases}\frac{1}{\sqrt{2}}, & u=0\\ 1, & u\neq 0\end{cases} \tag{3-64}$$

若 $f(x)(x=0,1,2,3,\cdots,N-1)$ 为 N 点离散序列,则一维 DCT 变换的定义为

$$F(u)=C(u)\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$
$$u=0,1,2,3,\cdots,N-1 \tag{3-65}$$

式中,$F(u)$是第 u 个余弦变换系数;u 是广义频率变量。

一维 DCT 逆变换定义为

$$f(x)=\sqrt{\frac{2}{N}}\sum_{u=0}^{N-1}C(u)F(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$
$$x=0,1,2,3,\cdots,N-1 \tag{3-66}$$

根据式(3-65)和式(3-66)可以看出,一维 DCT 正变换与逆变换的核相同。与 DFT 变换一样,DCT 变换也可以写成如下矩阵形式:

$$\boldsymbol{F}=\boldsymbol{G}\boldsymbol{f} \tag{3-67}$$

式中

$\boldsymbol{F}=[F(0),F(1),F(2),\cdots,F(N-1)]^{\mathrm{T}}$

$$\boldsymbol{G}=\frac{1}{\sqrt{N}}\begin{bmatrix}\sqrt{\frac{1}{N}}[1 & 1 & \cdots & 1]\\ \sqrt{\frac{2}{N}}\left[\cos\left(\frac{\pi}{2N}\right)\right. & \cos\left(\frac{3\pi}{2N}\right) & \cdots & \left.\cos\left(\frac{(2N-1)\pi}{2N}\right)\right]\\ \sqrt{\frac{2}{N}}\left[\cos\left(\frac{2\pi}{2N}\right)\right. & \cos\left(\frac{6\pi}{2N}\right) & \cdots & \left.\cos\left(\frac{(2N-1)2\pi}{2N}\right)\right]\\ \cdots & \cdots & \ddots & \cdots\\ \sqrt{\frac{2}{N}}\left[\cos\left(\frac{(N-1)\pi}{2N}\right)\right. & \cos\left(\frac{(N-1)3\pi}{2N}\right) & \cdots & \left.\cos\left(\frac{(N-1)(2N-1)\pi}{2N}\right)\right]\end{bmatrix}$$

$\boldsymbol{f}=[f(0),f(1),f(2),\cdots,f(N-1)]^{\mathrm{T}}$

3.2.2　二维 DCT 变换

一维 DCT 变换可以很方便地推广到二维 DCT 变换,设二维离散图像序列为$\{f(x,y)(x=0,1,\cdots,M-1;\ y=0,1,\cdots,N-1)\}$,则二维 DCT 正变换核为

$$g(x,y,u,v)=\sqrt{\frac{2}{MN}}C(u)C(v)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2x+1)v\pi}{2N}\right]$$

$$x,u=0,1,2,3,\cdots,M-1;\ y,v=0,1,2,3,\cdots,N-1 \tag{3-68}$$

其中,$C(u)$和 $C(v)$的定义与式(3-64)相同。

二维 DCT 正变换为

$$F(u,v)=\sqrt{\frac{2}{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)C(u)C(v)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2x+1)v\pi}{2N}\right]$$

$$u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1 \tag{3-69}$$

二维 DCT 逆变换定义形式为

$$f(x,y)=\sqrt{\frac{2}{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)C(u)C(v)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2x+1)v\pi}{2N}\right]$$

$$x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1 \tag{3-70}$$

与一维 DCT 变换一样,二维 DCT 变换也可以写成如下矩阵形式:

$$\boldsymbol{F}=\boldsymbol{GfG}^{\mathrm{T}} \tag{3-71}$$

根据式(3-69)和式(3-70)可以看出,二维 DCT 正变换与逆变换的核也相同,而且是可分离的,即可分离以后进行运算。

$$\begin{aligned}g(x,y,u,v)&=\sqrt{\frac{2}{MN}}C(u)C(v)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2x+1)v\pi}{2N}\right]\\&=\left\{\sqrt{\frac{2}{M}}C(u)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\right\}\left\{\sqrt{\frac{2}{N}}C(v)\cos\left[\frac{(2x+1)v\pi}{2N}\right]\right\}\\&=g_1(x,u)g_2(y,v)\end{aligned} \tag{3-72}$$

$$x,u=0,1,2,3,\cdots,M-1;\ y,v=0,1,2,3,\cdots,N-1$$

与 DFT 变换类似,根据可分离性原理,一次二维 DCT 变换可以通过二次一维 DCT 正变换完成。其算法流程如图 3-6 所示。

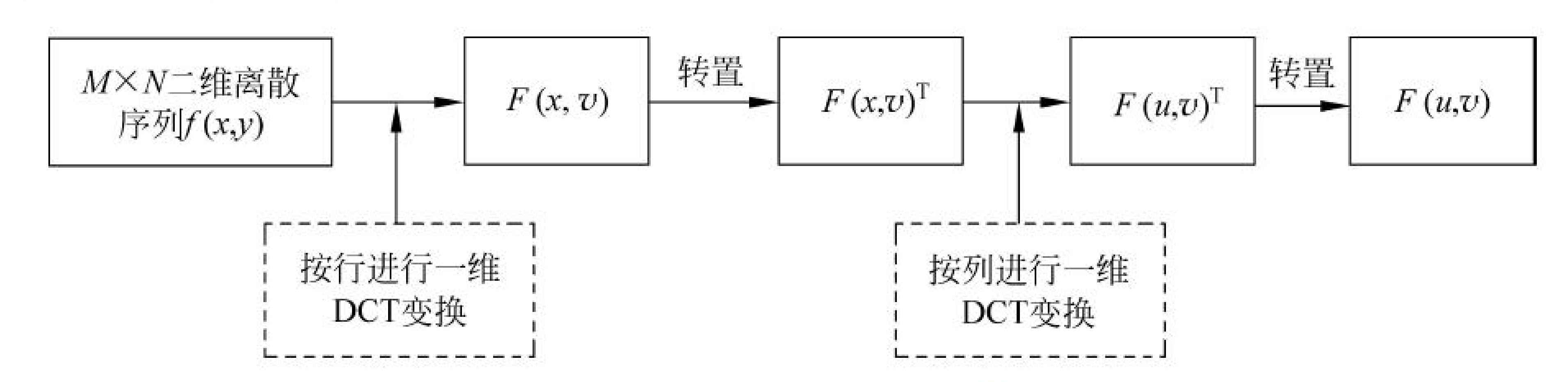


图 3-6　DCT 变换的分离运算流程

3.2.3　DCT 变换的快速算法

DCT 变换根据定义也可以直接进行计算,但计算量非常大,在实际应用中很不方便。目前,基于 DCT 的快速算法有许多种,由于 FFT 算法是一个应用广泛且非常成熟的快速算法,因此许多 DCT 算法也是基于 FFT 的原理建立起来的。具体步骤如下:

(1) 将离散序列 $f(x)$延拓为如下形式的 $2N$ 点序列。

$$f_1(x)=\begin{cases}f(x), & x=0,1,2,3,\cdots,N-1\\0, & x=N,N+1,\cdots,2N-1\end{cases} \tag{3-73}$$

(2) 根据一维 DCT 的定义,对延拓序列 $f_1(x)$进行 DCT 运算。

当 $u=0$ 时

$$F(0)=\sqrt{\frac{1}{N}}\sum_{x=0}^{N-1}f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{3-74}$$

当 $u=1,2,3,\cdots,N-1$ 时，有

$$\begin{aligned}
F(u)&=C(u)\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\\
&=\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]+\sqrt{\frac{2}{N}}\sum_{x=N}^{2N-1}0\cdot\cos\left[\frac{(2x+1)u\pi}{2N}\right]\\
&=\sqrt{\frac{2}{N}}\sum_{x=0}^{2N-1}f_1(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\\
&=\sqrt{\frac{2}{N}}\mathrm{Re}\left\{\sum_{x=0}^{2N-1}f_1(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\right\}\\
&=\sqrt{\frac{2}{N}}\mathrm{Re}\left\{\mathrm{e}^{-\mathrm{j}\pi\frac{u}{2N}}\sum_{x=0}^{2N-1}f_1(x)\mathrm{e}^{-\mathrm{j}2\pi\frac{ux}{2N}}\right\}\\
&=\sqrt{\frac{2}{N}}\mathrm{Re}\{\mathrm{e}^{-\mathrm{j}\pi\frac{u}{2N}}\mathrm{FFT}[f_1(x)]\}
\end{aligned}$$

$$(u=0,1,2,3,\cdots,N-1) \tag{3-75}$$

因此，DCT 快速算法可以通过对延拓序列 $f_1(x)$进行 FFT 运算完成，即将 N 点的序列 $f(x)$延拓为 $2N$ 点的序列 $f_1(x)$后，对序列 $f_1(x)$进行 FFT 运算，再将结果乘以 $\mathrm{e}^{-\mathrm{j}\pi\frac{u}{2N}}$并取其实部，然后乘以$\sqrt{\frac{2}{N}}$就是 DCT 运算的结果。

对于 DCT 逆变换(又称为 IDCT)，也可以采取类似的方法进行快速运算。

(1) 将离散序列 $F(u)$延拓为如下形式的 $2N$ 点序列。

$$F_1(u)=\begin{cases}F(x), & u=0,1,2,3,\cdots,N-1\\ 0, & u=N,N+1,\cdots,2N-1\end{cases}$$

(2) 根据一维 IDCT 的定义，对延拓序列 $F_1(u)$进行 IDCT 运算，公式为

$$f(x)=\left(\sqrt{\frac{1}{N}}-\sqrt{\frac{2}{N}}\right)F_1(0)+\sqrt{\frac{2}{N}}\mathrm{Re}\{\mathrm{IFFT}[F_1(u)]\mathrm{e}^{\mathrm{j}2\pi\frac{ux}{2N}}\} \tag{3-76}$$

3.2.4 二维 DCT 的频谱分布

以图像傅里叶变换为基础，二维 DCT 的频谱特性比较好理解，现以一个 DCT 变换实例来分析其频谱分布。

例 3-4：DCT 变换实例

图 3-7(a)为原始图像，图 3-7(c)为原始图像旋转 45°角的图像，图 3-7(b)和图 3-7(d)分别为图 3-7(a)和图 3-7(c)的 DCT 变换。对于 DCT，频率(0,0)点对应于频谱原点($u=0$，$v=0$)，低频在左上角，高频在右下角，$(M-1,N-1)$点对应于高频成分；DCT 低频系数值较大，高频系数值较小，即能量主要集中在低频，从左上角往右下角频率升高、能量降低。根据 DCT 变换原理，其运算相当于对带有中心偏移的实偶函数进行二维 DFT 运算，因此，DCT 的频谱分布与 DFT 相差一倍。

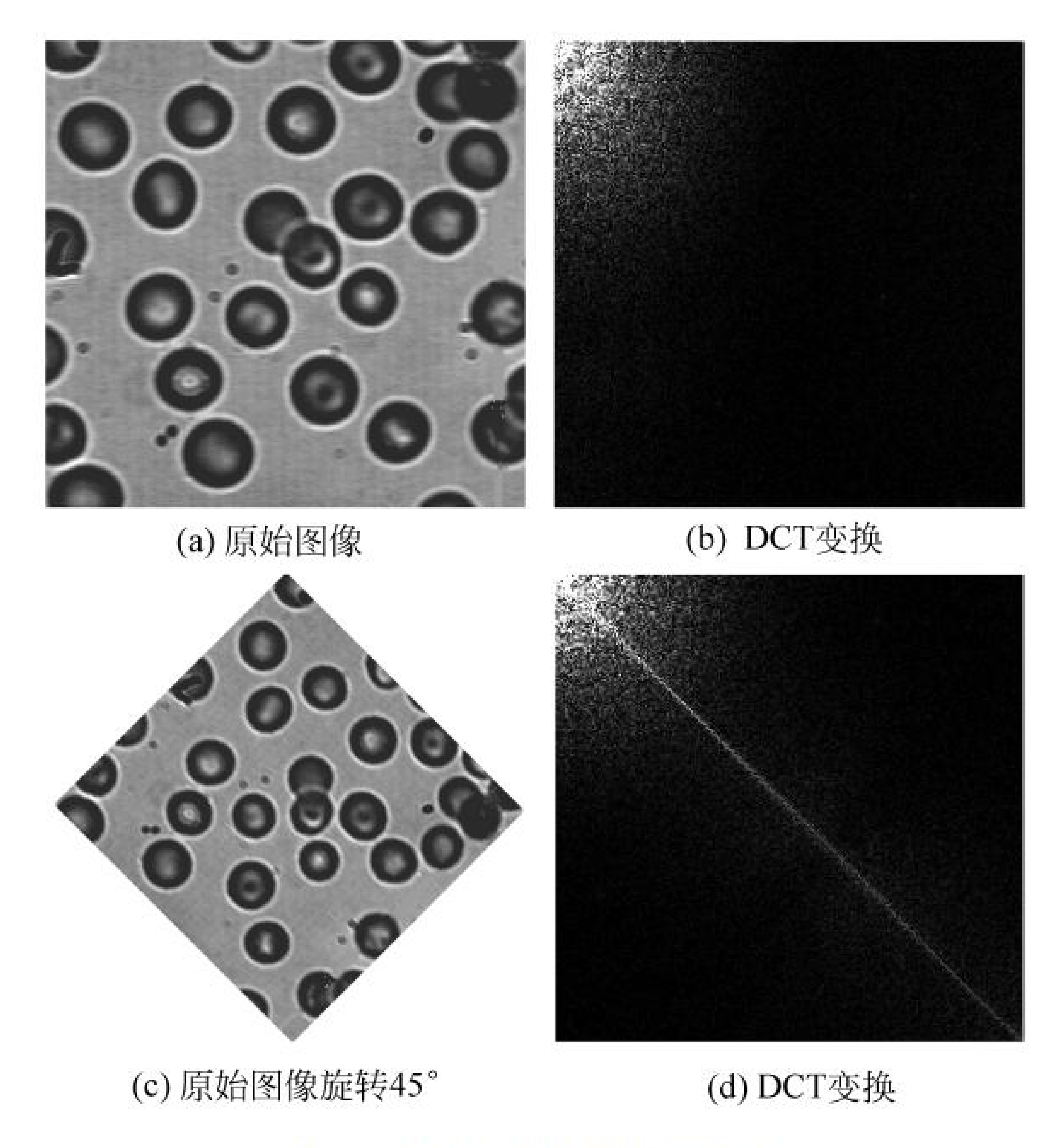


图 3-7　DCT 变换的频谱分布

3.3　离散 K-L 变换

K-L 变换(Karhunen-Loeve transform)是数字图像处理中具有广泛应用的一类重要变换,又称为特征向量变换、主分量变换或霍特林变换。K-L 变换既有连续形式的变换也有离散形式的变换,它是完全从图像的统计性质出发实现的变换。数字图像中应用的主要是离散 K-L 变换,它的重要优点是去相关性好,该变换在数据压缩、图像旋转、遥感多光谱图像的特征选择和统计识别等方面具有重要意义。

3.3.1　K-L 变换的基本原理

在实际应用中,二维图像可以视为随机场。一幅 $N\times N$ 的图像 $f(x,y)$ 在某个通信信道中传输了 M 次,由于受到随机噪声的干扰,接收到的可能是一个受干扰的图像随机变量的样本集合:

$$\{f_1(x,y),f_2(x,y),\cdots,f_M(x,y)\}$$

对第 i 次获得的图像 $f(x,y)$,可以用 $N^2\times 1$ 维向量 $\boldsymbol{X}_i$ 表示:

$$\boldsymbol{X}_i=[f_i(0,0),f_i(0,1),\cdots,f_i(0,N-1),f_i(1,0),f_i(1,1),\cdots,f_i(1,N-1),\cdots,$$
$$f_i(N-1,0),f_i(N-1,1),\cdots,f_i(N-1,N-1)]$$

若以 X_{ij} 表示第 i 次获得的图像 $f(x,y)$ 中的第 j 行的 N 个分量,则上式为

$$\boldsymbol{X}_i=[X_{i1},X_{i2},\cdots,X_{iN}]^{\mathrm{T}}$$

根据概率论和数理统计知识,图像随机场信号的随机变量之间的相关程度可以采用协方差矩阵表示,因此,向量 $\boldsymbol{X}$ 的协方差矩阵的定义为

$$\boldsymbol{C}_x = E[(\boldsymbol{X}-\boldsymbol{m}_x)(\boldsymbol{X}-\boldsymbol{m}_x)^{\mathrm{T}}] \tag{3-77}$$

式中，$\boldsymbol{C}_x$ 是 $N^2\times N^2$ 对称方阵，对角线上的元素 C_{nn} 表示第 n 个分量的方差，C_{nm} 表示第 n 个元素与第 m 个元素之间的协方差；E 表示数学期望运算；$\boldsymbol{m}_x$ 表示向量 $\boldsymbol{X}$ 的平均值。

向量 $\boldsymbol{X}$ 的平均值定义为

$$\boldsymbol{m}_x = E[\boldsymbol{X}] \tag{3-78}$$

对于 M 幅数字图像，向量 $\boldsymbol{X}$ 的均值 $\boldsymbol{m}_x$ 可以用全部样本的平均值近似计算，即

$$\boldsymbol{m}_x = E[\boldsymbol{X}] \approx \frac{\sum_{i=1}^{M}\boldsymbol{X}_i}{M} \tag{3-79}$$

向量 $\boldsymbol{X}$ 的协方差矩阵 $\boldsymbol{C}_x$ 可以采用同样的方法计算，即

$$\begin{aligned}\boldsymbol{C}_x &= E[(\boldsymbol{X}-\boldsymbol{m}_x)(\boldsymbol{X}-\boldsymbol{m}_x)^{\mathrm{T}}]\\ &\approx \frac{1}{M}\sum_{i=1}^{M}E[(\boldsymbol{X}-\boldsymbol{m}_x)(\boldsymbol{X}-\boldsymbol{m}_x)^{\mathrm{T}}]\\ &\approx \frac{1}{M}\left[\sum_{i=1}^{M}\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}\right]-\boldsymbol{m}_x\boldsymbol{m}_x^{\mathrm{T}}\end{aligned}$$

式中，$\boldsymbol{m}_x$ 是 N^2 个元素的向量。

$\boldsymbol{C}_x$ 是 $N^2\times N^2$ 实对称方阵，所以总可以求出协方差矩阵的 N^2 个特征值及其对应的正交特征向量。设特征值 $\lambda_i(i=1,2,\cdots,N^2)$ 对应的特征向量为 $e_i(i=1,2,\cdots,N^2)$，并将特征值按递减排序，即 $\lambda_1>\lambda_2>\cdots>\lambda_N^2$。则定义 K-L 变换矩阵 $\boldsymbol{A}$ 如下：

$$\boldsymbol{A} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1N^2}\\ e_{21} & e_{22} & \cdots & e_{2N^2}\\ \cdots & \cdots & \ddots & \cdots\\ e_{N^21} & e_{N^22} & \cdots & e_{N^2N^2}\end{bmatrix}$$

式中，e_{ij} 表示第 i 个特征向量的第 j 个分量。

于是，可得 K-L 变换形式为

$$\boldsymbol{Y} = \boldsymbol{A}(\boldsymbol{X}-\boldsymbol{m}_x) \tag{3-80}$$

向量 $\boldsymbol{Y}$ 的形式与 $\boldsymbol{X}$ 相同。该变换可直接理解为，由中心化图像向量 $(\boldsymbol{X}-\boldsymbol{m}_x)$ 与变换矩阵 $\boldsymbol{A}$ 相乘即可得到变换后的图像向量 $\boldsymbol{Y}$。

3.3.2 K-L 变换的性质

由于 K-L 变换的变换矩阵 $\boldsymbol{A}$ 是由协方差矩阵 $\boldsymbol{C}_x$ 的特征向量组成，因此，K-L 变换具有如下性质。

(1) 变换后图像向量 $\boldsymbol{Y}$ 的均值向量为 0(0 向量)，即

$$\boldsymbol{m}_y = E[\boldsymbol{Y}] = E[\boldsymbol{A}(\boldsymbol{X}-\boldsymbol{m}_x)] = \boldsymbol{A}E[\boldsymbol{X}]-\boldsymbol{A}\boldsymbol{m}_x = 0 \tag{3-81}$$

(2) 向量 $\boldsymbol{Y}$ 的协方差矩阵 $\boldsymbol{C}_y$ 可由 $\boldsymbol{A}$ 和 $\boldsymbol{C}_x$ 求得，公式为

$$\begin{aligned}\boldsymbol{C}_y &= E[(\boldsymbol{Y}-\boldsymbol{m}_y)(\boldsymbol{Y}-\boldsymbol{m}_y)^{\mathrm{T}}] = E\{[\boldsymbol{AX}-\boldsymbol{A}\boldsymbol{m}_x][\boldsymbol{AX}-\boldsymbol{A}\boldsymbol{m}_x]^{\mathrm{T}}\}\\ &= E[\boldsymbol{A}(\boldsymbol{X}-\boldsymbol{m}_x)(\boldsymbol{X}-\boldsymbol{m}_x)^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}]\\ &= \boldsymbol{A}E[(\boldsymbol{X}-\boldsymbol{m}_x)(\boldsymbol{X}-\boldsymbol{m}_x)^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\\ &= \boldsymbol{A}\boldsymbol{C}_x\boldsymbol{A}^{\mathrm{T}}\end{aligned} \tag{3-82}$$

(3) 经过 K-L 变换后完全消除了元素之间的相关性，即协方差矩阵 $\boldsymbol{C}_y$ 是对角矩阵，且主对角线上的元素等于 $\boldsymbol{C}_x$ 的特征值，即

$$\boldsymbol{C}_y = \begin{bmatrix} \lambda_1 & & & & & & 0 \\ & \lambda_2 & & & & & \\ & & \lambda_3 & & & & \\ & & & \ddots & & & \\ & & & & \lambda_i & & \\ & & & & & \ddots & \\ 0 & & & & & & \lambda_{N^2} \end{bmatrix} \tag{3-83}$$

协方差矩阵 $\boldsymbol{C}_y$ 主对角线以外的元素全部为 0，即变换图像向量 $\boldsymbol{Y}$ 的各个元素是互不相关的，由于 $\lambda_i(i=1,2,\cdots,N^2)$ 也是 $\boldsymbol{C}_x$ 的特征值，所以 $\boldsymbol{C}_y$ 和 $\boldsymbol{C}_x$ 具有完全相同的特征值和特征向量。

3.3.3 K-L 变换的逆变换

与其他变换类似，K-L 变换也有逆变换，即可以通过 $\boldsymbol{Y}$ 来重建 $\boldsymbol{X}$。由于矩阵 $\boldsymbol{A}$ 的各行都是正交归一化矢量，所以有

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\mathrm{T}}$$

根据 K-L 变换形式可得

$$\boldsymbol{X} = \boldsymbol{A}^{-1}\boldsymbol{Y} + \boldsymbol{m}_x \tag{3-84}$$

由于协方差矩阵特征值是由大到小排列的，在许多实际应用中，可以充分利用 K-L 变换的这个性质进行近似计算。例如，可以只取一部分特征值及其对应的特征向量，若取前 n 个特征值，则可以表示为

$$\boldsymbol{C}_x = \boldsymbol{C}_y = \begin{bmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & \ddots & \\ 0 & & & & \lambda_n \end{bmatrix}$$

即可以只取前 n 个分量重建 X 的近似值

$$\boldsymbol{X}_n = \boldsymbol{A}_n^{-1}\boldsymbol{Y} + \boldsymbol{m}_x \tag{3-85}$$

因此，可以得出 $\boldsymbol{X}_n$ 和 $\boldsymbol{X}$ 之间的均方误差为

$$\sigma_n = \sum_{k=n+1}^{N^2} \lambda_k \tag{3-86}$$

显然，若 $k=N^2$，则均方误差为 0，即可以精确地重构 X。由于特征值 $\lambda_i(i=1,2,\cdots,N^2)$ 是单调递减的，因此，可以通过取不同的 n 值来控制 X_n 和 X 之间的均方误差，即完全可以根据误差的要求来控制所取特征值的数量。因此，理论上 K-L 变换可以将均方误差控制到任意小的程度，或者说可以实现均方误差最小意义下的最优变换。

综上所述，K-L 变换具有许多优点，其最重要的优点是去相关性能非常好，它可以完全解除数据之间的相关性，因此，K-L 变换可广泛用于图像数据的旋转或压缩处理。但 K-L 变换也具有自身的明显缺点，这就是二维 K-L 变换不可分离的变换，不能通过在 x 与 y 方

向上的两次一维的 K-L 变换来完成二维 K-L 变换的运算。同时，K-L 变换是一种和图像数据有关的变换，在变换中，必须计算图像数据的 $N^2 \times N^2$ 协方差矩阵的特征值和特征向量，计算量非常庞大，因而 *K-L* 变换在实际图像处理中并没有得到所期望的普及程度。

3.4 离散沃尔什变换

DFT 或 DCT 变换都是以正弦或余弦三角函数为基本的正交函数基，而 DFT 的快速运算是复数范围内进行运算，不仅运算量庞大，而且运算复杂，DCT 变换虽然避免了复数运算，但需要进行三角函数运算，运算复杂程度依然很高。因此，无论 DFT，还是 DCT 运算，占用时间仍然较多，而在实际应用中，有时需要更为有效和便利的变换方法。沃尔什变换就是其中一种，它有效地避免了因数据本身原因所产生的运算复杂性，它由只有＋1 和－1 两个数值所构成的完备正交基组成。由于沃尔什函数基是二值正交基，与数字逻辑的两个状态相对应，因而更加适用于计算机技术、数字信号处理等应用领域。

此外，与 DFT 相比，沃尔什变换减少了存储空间并提高了运算速度，这一点对图像处理来说是至关重要的，特别是在大量数据需要进行实时处理时，沃尔什变换更加显示出其优越性。

3.4.1 一维离散沃尔什变换

沃尔什(Walsh)变换与沃尔什函数密切相关，沃尔什函数是 1923 年由美国数学家沃尔什首先提出的。沃尔什函数系是一个完备正交函数系，其值只能取＋1 和－1。从排列次序上可将沃尔什函数分为三种定义方法：一是按照佩利排列来定义(按自然排序)；二是按照沃尔什排列来定义(按列率排序)；三是按照哈达玛排列来定义，又称为哈达玛变换。

1. 沃尔什正变换

设 $f(x)$表示 N 点的一维离散序列，则一维沃尔什变换定义为

$$F(u) = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_{n-1-i}(u)}, \quad (u = 0,1,2,3,\cdots,N-1) \tag{3-87}$$

其中，一维沃尔什变换核为

$$g(x,u) = \frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_{n-1-i}(u)} \tag{3-88}$$

式中，$u=0,1,2,3,\cdots,N-1$；$x=0,1,2,3,\cdots,N-1$。N 是沃尔什变换的阶数，$N=2^n$。$b_i(z)$是 z 的二进制数的第 i 位数值，取值为 0 或 1。如 $i=6$，由于 6 的二进制表示为 110，因此 $b_0(z)=0$，$b_1(z)=1$，$b_2(z)=1$。

2. 沃尔什逆变换

一维离散沃尔什逆变换定义为

$$f(x) = \frac{1}{\sqrt{N}}\sum_{u=0}^{N-1} F(u)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_{n-1-i}(u)}, \quad x = 0,1,2,3,\cdots,N-1 \tag{3-89}$$

逆变换的核为

$$h(x,u) = g(x,u)$$

一维沃尔什正、反变换核相同，沃尔什变换核是一个对称阵列，其行和列是正交的。沃

尔什正、反变换形式本质上相同，因此，计算沃尔什变换的算法可直接用来求其逆变换。一维沃尔什变换也具有快速算法，简称为 FWT，在形式上和 FFT 算法类似。当 $N=8$ 时，其变换核用矩阵表示为

$$\boldsymbol{G}=\frac{1}{\sqrt{8}}\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&-1&-1&-1&-1\\1&1&-1&-1&1&1&-1&-1\\1&1&-1&-1&-1&-1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&-1&1&-1&-1&1&-1&1\\1&-1&-1&1&1&-1&-1&1\\1&-1&-1&1&-1&1&1&-1\end{bmatrix}$$

3.4.2 二维离散沃尔什变换

1. 二维沃尔什正变换

设 $f(x,y)$ 表示 $M\times N$ 的二维离散序列，则二维沃尔什变换定义为

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)g(x,u,y,v),\quad (u=0,1,2,3,\cdots,M-1;\ v=0,1,2,3,\cdots,N-1) \tag{3-90}$$

其中，二维离散沃尔什变换核为

$$g(x,u,y,v)=\frac{1}{\sqrt{MN}}\sum_{u=0}^{N-1}(-1)^{\sum_{i=0}^{n-1}[b_i(x)b_{n-1-i}(u)]+\sum_{j=0}^{n-1}[b_j(x)b_{n-1-j}(u)]} \tag{3-91}$$

式中，$M=2^m$，$N=2^n$。

二维离散沃尔什变换的变换核是可分离的，即

$$\begin{aligned}g(x,u,y,v)&=\frac{1}{\sqrt{MN}}\sum_{u=0}^{N-1}(-1)^{\sum_{i=0}^{m-1}[b_i(x)b_{n-1-i}(u)]+\sum_{j=0}^{n-1}[b_j(x)b_{n-1-j}(u)]}\\&=\frac{1}{\sqrt{M}}\sum_{u=0}^{N-1}(-1)^{\sum_{i=0}^{m-1}[b_i(x)b_{n-1-i}(u)]}\ \frac{1}{\sqrt{N}}\sum_{u=0}^{N-1}(-1)^{\sum_{j=0}^{n-1}[b_j(x)b_{n-1-j}(u)]}\\&=g_1(x,u)g_2(y,v)\end{aligned} \tag{3-92}$$

根据沃尔什变换的定义形式可以得出，二维沃尔什变换具有可分离特性，即一次二维沃尔什变换可以通过两次一维沃尔什变换来实现。

2. 二维沃尔什逆变换

二维沃尔什逆变换定义为

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)h(x,u,y,v),\quad x=0,1,2,3,\cdots,M-1;\ y=0,1,2,3,\cdots,N-1 \tag{3-93}$$

二维离散沃尔什逆变换核为

$$g(x,u,y,v)=\frac{1}{\sqrt{MN}}\sum_{u=0}^{N-1}(-1)^{\sum_{i=0}^{m-1}[b_i(x)b_{n-1-i}(u)]+\sum_{j=0}^{n-1}[b_j(x)b_{n-1-j}(u)]} \tag{3-94}$$

式中，$M=2^m$，$N=2^n$。

二维沃尔什逆变换的核为

$$h(x,u,y,v)=g(x,u,y,v)$$

同样，二维逆变换具有可分离性。二维沃尔什变换也可以表示为矩阵形式，即

$$\begin{cases} \boldsymbol{F}=\dfrac{1}{\sqrt{MN}}\boldsymbol{G}_1\boldsymbol{f}\boldsymbol{G}_2 \\ \boldsymbol{f}=\dfrac{1}{\sqrt{MN}}\boldsymbol{G}_1\boldsymbol{F}\boldsymbol{G}_2 \end{cases} \tag{3-95}$$

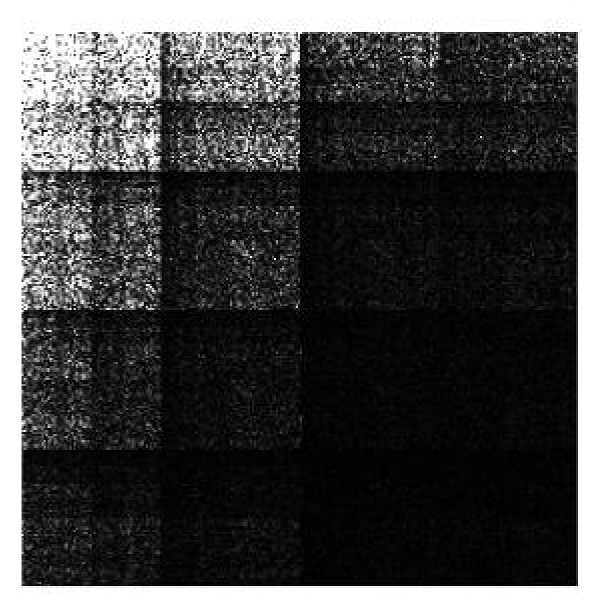
式中，$\boldsymbol{G}_1$ 为 $M\times M$ 变换核方阵；$\boldsymbol{G}_2$ 为 $N\times N$ 变换核方阵。

3.4.3　沃尔什变换的频谱

和 DCT 变换相同，二维沃尔什变换也具有能量集中的性质，原始图像数据越是均匀分布，沃尔什变换后的数据越集中于矩阵的边角上，因此，二维沃尔什变换也常用于压缩图像信息。

例 3-5：沃尔什变换实例

如图 3-8 所示是沃尔什变换的实例，图 3-8(a)为对应于图 3-7(a)的沃尔什变换结果，图 3-8(b)为对应于图 3-7(c)的沃尔什变换结果。



(a) 原始图像的沃尔什变换　　(b) 旋转45° 沃尔什变换

图 3-8　沃尔什变换的频谱分布

沃尔什变换是将一个函数变换成取值为+1或−1的基本函数构成的级数，用它来逼近数字脉冲信号时要比 DFT 有利。因此，它在图像传输、通信技术和数据压缩中获得了广泛的使用。同时，沃尔什变换是实数，所以对工程应用问题，沃尔什变换的存储量比 DFT 少，而且运算速度非常快。

例 3-6：有两个二维数字图像信号，即

(1) $f_1=\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$

(2) $f_2=\begin{bmatrix}1&3&3&1\\1&3&3&1\\1&3&3&1\\1&3&3&1\end{bmatrix}$

分别求 f_1 和 f_2 的二维沃尔什变换。

离散二维沃尔什变换的求解过程如下：

根据理论，对于(1)和(2)，均有 $M=N=4$，因此，其二维沃尔什变换的核为

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

由此可得

$$F_1(u)=\frac{1}{4^2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\times\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\times\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$F_2(u)=\frac{1}{4^2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\times\begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix}\times\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$=\begin{bmatrix} 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3.5 离散哈达玛变换

哈达玛(Hadamard)变换与沃尔什变换十分类似，就其本质而言，哈达玛变换是一种特殊排序的沃尔什变换。因此，有的书上称为沃尔什-哈达玛变换。哈达玛变换矩阵也是一个仅包括+1和−1两个矩阵元素的方阵，任意两行或两列相乘后的各数之和必定为零，即不同的行或不同的列之间都彼此正交，哈达玛变换核矩阵与沃尔什变换不同之处仅仅是行的次序不同。哈达玛变换的最大优点在于它的变换核矩阵具有简单的递推关系，即高阶矩阵可以通过低阶矩阵求出。因此，许多人基于这个特点更愿意应用哈达玛变换。

3.5.1 一维离散哈达玛变换

1. 一维哈达玛正变换

设 $f(x)$表示 N 点的一维离散序列，则一维哈达玛变换定义为

$$F(u)=\sum_{x=0}^{N-1}f(x)g(x,u)=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}f(x)(-1)^{\sum_{i=0}^{n-1}b_i(x)b_i(u)},\quad (u=0,1,2,3,\cdots,N-1) \tag{3-96}$$

其中，$g(x,u)$是一维哈达玛变换的核，定义为

$$g(x,u)=\frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{N-1}b_i(x)b_i(u)} \tag{3-97}$$

式中，$u=0,1,2,3,\cdots,N-1$；$x=0,1,2,3,\cdots,N-1$。N 是哈达玛变换的阶数，$N=2^n$。$b_i(z)$是 z 的二进制数的第 i 位数值，取值为 0 或 1。

2. 一维哈达玛逆变换

若已知 N 点的一维离散序列 $F(u)$，则可以进行哈达玛逆变换，其定义为

$$f(x)=\sum_{u=0}^{N-1}F(u)h(x,u)=\frac{1}{\sqrt{N}}\sum_{u=0}^{N-1}F(u)(-1)^{\sum_{i=0}^{n-1}b_i(x)b_i(u)},\quad (x=0,1,2,3,\cdots,N-1) \tag{3-98}$$

与正变换相同，$h(x,u)$是一维哈达玛逆变换的核，逆变换核与正变换核相等，即

$$h(x,u)=g(x,u)=\frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{N-1}b_i(x)b_i(u)} \tag{3-99}$$

哈达玛变换的阶数具有规律性，即按照 $N=2^n$ 规律递升，高阶哈达玛矩阵可以通过低阶哈达玛矩阵的克罗尼科积运算求得。为了表示方便，一般将哈达玛变换系数$\left(\frac{1}{\sqrt{N}}\right)$作为前置系数，不同阶的哈达玛矩阵具有如下关系：

(1) $\boldsymbol{H}_1=[1]$

(2) $\boldsymbol{H}_2=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$

(3) $\boldsymbol{H}_4=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$

(4) $\boldsymbol{H}_N=\boldsymbol{H}_{2^n}=\boldsymbol{H}_2\otimes\boldsymbol{H}_{\frac{N}{2}}=\begin{bmatrix}\boldsymbol{H}_{\frac{N}{2}} & \boldsymbol{H}_{\frac{N}{2}}\\ \boldsymbol{H}_{\frac{N}{2}} & -\boldsymbol{H}_{\frac{N}{2}}\end{bmatrix}$

采用上述规律求哈达玛变换矩阵要比直接用哈达玛变换核求矩阵快得多，此结论提供了一种快速哈达玛变换，也可称为 FHT。例如，根据哈达玛矩阵的运算规律，可以得出 8 阶哈达玛矩阵为

$$\boldsymbol{H}_8=\begin{bmatrix}\boldsymbol{H}_4 & \boldsymbol{H}_4\\ \boldsymbol{H}_4 & -\boldsymbol{H}_4\end{bmatrix}$$

$$=\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\end{bmatrix}$$

还有一种常用的哈达玛变换称为定序哈达玛变换，定序的哈达玛变换是由前面介绍的哈达玛变换演变而来。在哈达玛变换矩阵中，通常将某一列元素符号改变的总次数称为这个列的列率。则前面给出的 $N=8$ 时的变换矩阵 $\boldsymbol{H}_8$ 的 8 个列的列率分别为 0、7、3、4、1、6、2、5。而下面要介绍的定序哈达玛变换的变换矩阵的列率是随 u 的增加而递增的。例如 $N=8$ 时，定序哈达玛变换矩阵的列率从第 1 列到第 8 列分别为 0、1、2、3、4、5、6、7。

当 $N=2^n$ 时，定序哈达玛正变换核和逆变换核相同，其变换核为

$$g(x,u)=h(x,u)=\frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{N-1}b_i(x)p_i(u)} \tag{3-100}$$

$p_i(u)$按以下递推关系求得

$$\begin{aligned}
p_0(u)&=b_{n-1}(u)\\
p_1(u)&=b_{n-1}(u)+b_{n-2}(u)\\
p_2(u)&=b_{n-2}(u)+b_{n-3}(u)\\
&\vdots\\
p_{n-1}(u)&=b_1(u)+b_0(u)
\end{aligned}$$

因此，定序的哈达玛正反变换对为

$$F(u)=\sum_{x=0}^{N-1}f(x)g(x,u)=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}f(x)(-1)^{\sum_{i=0}^{n-1}b_i(x)p_i(u)} \tag{3-101}$$

$$f(x)=\sum_{u=0}^{N-1}F(u)h(x,u)=\frac{1}{\sqrt{N}}\sum_{u=0}^{N-1}F(u)(-1)^{\sum_{i=0}^{n-1}b_i(x)p_i(u)} \tag{3-102}$$

例 3-7：根据定序的哈达玛变换核，求 8 阶哈达玛变换矩阵。

由于定序哈达玛变换的列率是递增规律，因此，8 阶哈达玛变换矩阵为

$$\boldsymbol{H}_8=\frac{1}{\sqrt{8}}\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1
\end{bmatrix}$$

3.5.2 二维离散哈达玛变换

一维哈达玛变换可以很方便地推广到二维，其正变换定义为

$$\begin{aligned}
F(u,v)&=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)g(x,u,y,v)\\
&=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)(-1)^{\sum_{i=0}^{m-1}b_i(x)b_i(u)+\sum_{i=0}^{n-1}b_i(y)b_i(v)}
\end{aligned} \tag{3-103}$$

逆变换为

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)h(x,u,y,v)$$

$$=\frac{1}{\sqrt{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)(-1)^{\sum_{i=0}^{m-1}b_i(x)b_i(u)+\sum_{i=0}^{n-1}b_i(y)b_i(v)} \tag{3-104}$$

二维哈达玛正变换的核为

$$g(x,u,y,v)=\frac{1}{\sqrt{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}(-1)^{\sum_{i=0}^{m-1}b_i(x)b_i(u)+\sum_{i=0}^{n-1}b_i(y)b_i(v)} \tag{3-105}$$

其中，$x,u=0,1,2,3,\cdots,M-1$；$y,v=0,1,2,3,\cdots,N-1$。

二维哈达玛逆变换核与正变换核相等，即

$$h(x,u,y,v)=g(x,u,y,v) \tag{3-106}$$

二维哈达玛变换核是可分离和对称的，因此一次二维哈达玛变换也可分为两次一维哈达玛变换的计算而实现。二维哈达玛变换也有相应的定序的哈达玛变换，与一维情况类似，只需将二维正反变换定义中的 $b_i(u)$ 或 $b_i(v)$ 改为对应的 $p_i(u)$ 或 $p_i(v)$ 即可。

根据运算原理，哈达玛变换矩阵具有简单的递推关系，且正、反变换矩阵完全相同，只包含实数的加、减法运算而没有复数的乘法运算，使得计算速度快、存储空间少，有利于硬件实现，对实时处理和大量数据操作具有特殊吸引力，因此获得了广泛的应用。如通信领域中的多路数字通信系统、语音加密、视频编码系统、雷达系统、图像通信系统；在信号处理领域中的信号分析与综合、功率谱分析、模式识别、图像处理。特别是在图像传输、存储系统中，用于图像压缩非常有效。该变换虽然具有上述许多优点，但与建立在正弦函数基础上的傅里叶变换相比，哈达玛变换在理论上和实践上还有一些问题需要进一步的研究。如相关与卷积的运算，如何从经济上和技术上解决以矩形波为基础的设备，来取代现有以正弦波为基础的大量设备等问题。

3.6 小波变换

小波变换近年来在图像处理中受到了前所未有的重视，面向图像压缩、特征检测以及纹理分析的许多新方法，如多分辨率分析、时频域分析、金字塔算法等，都可以归于小波变换(wavelet transforms)的范畴。与傅里叶变换相比，小波变换由于其在高频时具有的时间精度和低频时所具有的频率精度，能自动适应时频信号分析的要求，可以聚焦到信号的任意细节等显著特点而得到了越来越广泛的研究和重视，小波变换被迅速应用到图像和语音分析等众多领域。近十年来，对小波变换理论研究已成为应用数学和信号处理领域的一个新方向。

3.6.1 小波变换简介

小波变换是法国从事地质勘探工作的信号处理工程师 J. Morlet 于 1974 年首先提出的。Morlet 根据物理及信号处理的应用需求提出了推演公式，但在当时的条件下，如同欧拉提出任一函数都能展开成三角函数的无穷级数一样，未能得到著名数学家 Lagrange、Laplace 等的认可。

实际上，在 20 世纪 70 年代，著名科学家 A. Calderon 表示定理的发现以及 Hardy 空间的原子分解和无条件基的深入研究为小波变换的诞生做了理论上的准备，著名数学家 Y. Meyer于 1986 年第一次构造出一个真正的小波基，并与 S. Mallat 合作建立了构造小波

基的方法及多尺度分析法，随后，小波变换开始受到广泛的重视。其中，比利时女数学家 I. Daubechies撰写的 *Ten Lectures on Wavelets* 对小波的普及起了重要的推动作用。

与傅里叶变换、加伯变换(加窗傅里叶变换)相比，小波变换又被称为"数学显微镜"，是一个时间和频率的局域变换，因而能有效地从信号中提取信息，通过伸缩和平移等运算对信号进行多尺度细化分析，有效解决了傅里叶变换面临的许多问题，成为继傅里叶变换以来在科学方法上的重大突破之一。

小波变换具有广泛的适应性，特别是对非平稳随机信号，小波变换同样可以适应。作为一种数学分析工具，小波变换可以用于许多学科，如信号分析、量子力学、理论物理、数值分析、曲线曲面构造、微分方程求解、控制论、军事电子对抗、军械装备智能化等；小波分析在信号处理中的应用也十分广泛，它可以用于信号处理的各方面，如信号滤波、时频分析、信噪分离、弱信号提取、求分形指数、信号识别与诊断和多尺度边缘检测等。小波分析还可用于图像滤波、去噪声、图像压缩等很多方面，基于小波的图像压缩算法不仅压缩比高、速度快，而且压缩后能保持信号与图像的特征不变，在传输中具有一定的抗干扰性。

3.6.2 连续小波变换

傅里叶变换是以两端无限延伸的正弦波作为基函数的，其重要特点是可以展现信号的频域特征，但却基本消失了信号的时间局部化信息。为了克服傅里叶变换的不足，人们提出了小波变换。小波变换的基函数在频率上和位置上都是变化的，是有限宽度的波，称为小波(wavelet)，基于它们的变换就是小波变换。小波变换能同时刻画信号时频两域的特性，它通过放缩母小波来获得信号的频率信息，通过平移母小波来获得信号的时间信息。对小波的放缩和平移是计算小波系数的需要，小波系数反映了小波和局部信号之间的相关特性。

1. 一维连续小波变换

设函数 $f(t)$具有有限能量，即 $f(t)\in L^2(R)$，则连续小波变换(CWT)的定义为

$$W_f(a,b) = < f,\psi_{a,b}(x) > = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t)\mathrm{d}t$$

$$= \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(t)\,\overline{\psi\left(\frac{t-b}{a}\right)},\quad a>0, f(t)\in L^2(R) \tag{3-107}$$

式中，a 为尺度因子；b 为位移因子；函数 $\psi_{a,b}(t)$称为小波。

连续小波变换也称为积分小波变换，积分核为 $\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty}\psi\left(\frac{t-b}{a}\right)\mathrm{d}t$ 的函数族。若 $a>1$ 函数 $\psi(t)$具有伸展作用，若 $a<1$ 函数 $\psi(t)$具有收缩作用。随着 a 的减小，$\psi_{a,b}(\omega)$的支撑区间也随之变窄，而 $\psi_{a,b}(\omega)$的频谱随之向高频展宽，反之亦然。因此，小波变换可以实现窗口大小的自适应变化。当信号频率升高时，时窗宽度变窄，而频窗宽度则增大，从而有利于提高时域分辨力；当信号频率降低时，时窗宽度增大，而频窗宽度则变窄，提高频率分辨力。

2. 小波的选择

小波 $\psi(t)$的选择不是唯一的，但也不是任意的，$\psi(t)$是具有归一化、具有单位能量的解析函数，所有小波是通过对基本小波进行尺度伸缩和位移得到的。

(1) 基本小波是一具有特殊性质的函数，它是振荡衰减的，而且通常衰减得很快，在数

学上满足积分为零的条件：

$$\int_{-\infty}^{+\infty} \psi(t)\mathrm{d}t = 0 \tag{3-108}$$

因此，基本小波是一个积分为零且能量集中在以 $t=0$ 为中心的邻域内，而且小波函数还具有速降性和紧支性。其高阶矩也为 0，即

$$\int_{-\infty}^{+\infty} t^k \psi(t)\mathrm{d}t = 0, \quad k = 0,1,2,3,\cdots,N-1 \tag{3-109}$$

该条件称为小波的容许条件：

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\psi(\omega)|^2}{\omega}\mathrm{d}\omega < \infty \tag{3-110}$$

由于 $\psi(\omega) = \int_{-\infty}^{+\infty} \psi(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$，$C_\psi < \infty$，因此，$\psi(\omega)$连续可积，即

$$\psi(0) = \int_{-\infty}^{+\infty} \psi(t)\mathrm{d}t = 0 \tag{3-111}$$

根据式(3-111)可以得出，小波 $\psi(t)$在 t 轴上具有正负取值才可能满足上式的积分为 0，因此，$\psi(t)$应具有振荡性。

(2) 小波 $\psi(t)$的定义域是紧支撑的，即在一个很小的区间之外，小波 $\psi(t)$迅速衰减为 0，也就是说小波函数 $\psi(t)$具有速降性。

综上所述，小波 $\psi(t)$是一种具有振荡性的且迅速衰减的波。

3. 小波变换的逆变换

对于所有的 $f(t)$，$\psi(t) \in L^2(R)$，则连续小波变换的逆变换定义为

$$f(t) = \frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} a^{-2} W_f(a,b)\psi_{a,b}(t)\mathrm{d}a\mathrm{d}b \tag{3-112}$$

4. 二维连续小波变换

二维小波变换也分为二维连续小波变换和二维离散小波变换。

(1) 二维连续小波变换为

$$W_f(a,b_x,b_y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\psi_{a,b_x,b_y}(x,y)\mathrm{d}x\mathrm{d}y \tag{3-113}$$

式中，$\psi_{a,b_x,b_y}(x,y)$为二维连续小波基函数，即

$$\psi_{a,b_x,b_y}(x,y) = \frac{1}{|a|}\psi\left(\frac{y-b_x}{a},\frac{y-b_y}{a}\right) \tag{3-114}$$

(2) 二维小波变换的逆变换为

$$f(x,y) = \frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} a^{-3} W_f(a,b_x,b_y)\psi_{a,b_x,b_y}(x,y)\mathrm{d}b_x\mathrm{d}b_y\mathrm{d}a \tag{3-115}$$

二维连续小波的允许条件如下：

(1) 定义域为紧支撑集。

(2) 均值为 0，即

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \psi(x,y)\mathrm{d}x\mathrm{d}y = 0$$

5. 几种典型的小波

小波 $\psi(t)$的选择是非常灵活的，凡满足条件的函数均可以作为小波函数，其中 Haar 小波、Mexico Hat 小波和 Morlet 小波是常用的小波函数。

1）Haar 小波

Haar 小波函数为

$$\psi_H(t)=\begin{cases}1, & 0\leqslant t<\dfrac{1}{2}\\ -1, & \dfrac{1}{2}\leqslant t<1\\ 0, & 其他\end{cases} \tag{3-116}$$

该正交函数是 Haar 于 1990 年提出的函数，对 t 平移可得到

$$\int_{-\infty}^{+\infty}\psi_H(t)\psi_{a,b}(t-n)\mathrm{d}t=0,\quad n=0,\pm 1,\pm 2,\pm 3,\cdots \tag{3-117}$$

Haar 函数波形如图 3-9 所示。

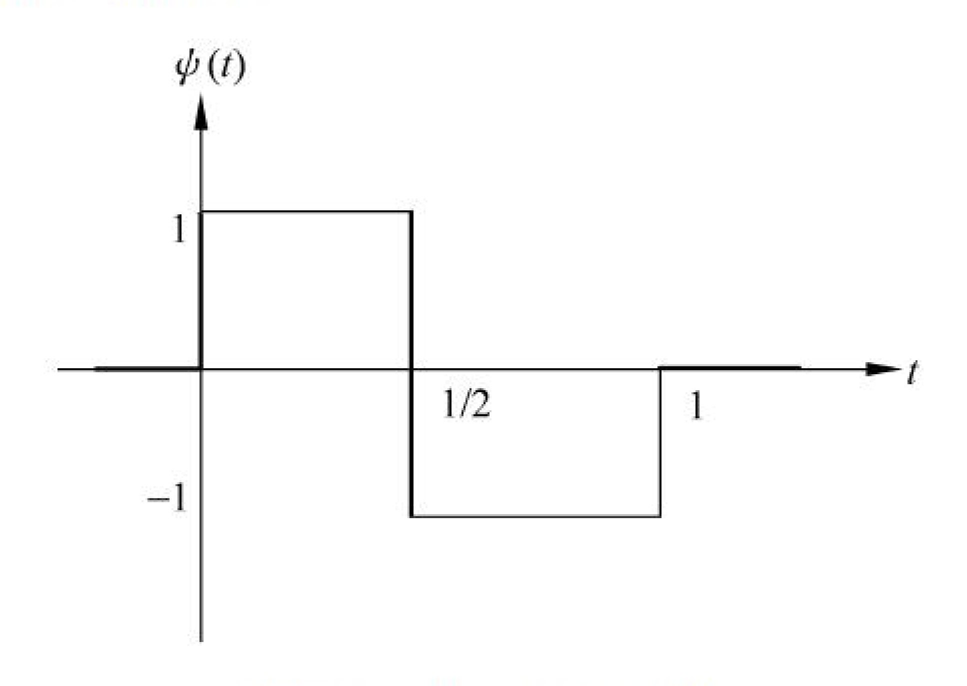


图 3-9 Haar 小波波形

如图 3-10 所示为两个小波 $\psi_{1,0}(t)$ 和 $\psi_{1,1}(t)$ 的波形。

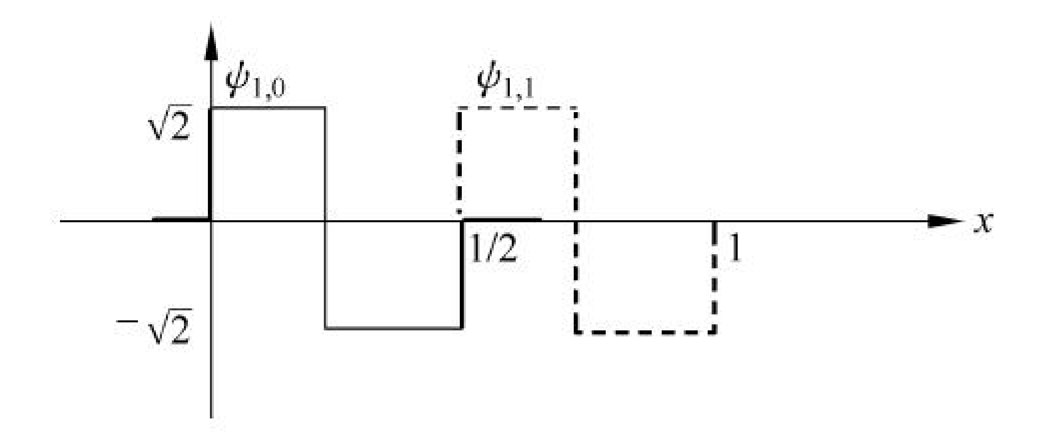


图 3-10 两个 Haar 小波

该基本小波定义的小波变换称为 Haar 小波变换，是各种常用的小波变换中最简单的一种变换形式。

2）Mexico Hat 小波

Mexico Hat 小波是 Gauss 函数的二阶导数，其函数形式为

$$\psi(t)=\frac{2}{\sqrt{3}}\pi^{-\frac{1}{4}(1-t^2)}\mathrm{e}^{-\frac{t^2}{2}} \tag{3-118}$$

Mexico Hat 小波也称为 Marr 小波，Mexico Hat 小波是实数函数小波，它的更一般形式由 Gauss 函数的 n 阶导数定义，即

$$\psi_n(t)=(-1)^n\frac{\mathrm{d}^n}{\mathrm{d}t^n}(\mathrm{e}^{-\frac{|x|^2}{2}}) \tag{3-119}$$

相应的谱为

$$\psi_n(\omega) = n(\mathrm{j}\omega)^n \mathrm{e}^{-\frac{|\omega|^2}{2}} \tag{3-120}$$

Mexico Hat 小波的波形如图 3-11 所示。

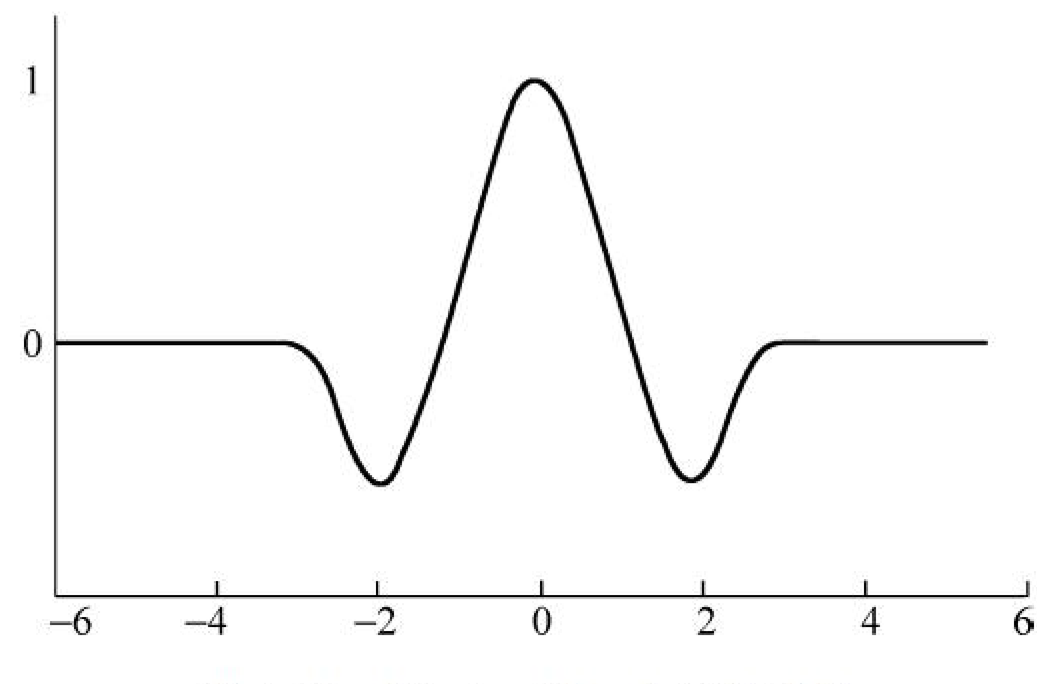


图 3-11 Mexico Hat 小波的波形

3）Morlet 小波

Morlet 小波是最常用的复数值小波函数，其形式为

$$\psi(t) = -\pi^{-\frac{1}{4}}\left[\mathrm{e}^{-\mathrm{j}\omega_0 t} - \mathrm{e}^{-\frac{\omega_0^2}{2}}\right]\mathrm{e}^{-\frac{t^2}{2}} \tag{3-121}$$

6. 小波变换的性质

小波变换常用的重要特性如下：

（1）小波变换是一个满足能量守恒方程的线性运算，它将信号分解成对空间和尺度（即时间和频率）的独立贡献，同时又不失原信号所包含的信息。

（2）小波变换不一定要求是正交的，小波基不唯一。小波函数系的时宽-带宽积很小，且在时间和频率轴上都很集中，即展开系数的能量很集中。

（3）小波变换相当于一个具有放大、缩小和平移等功能的数学显微镜，通过检查不同放大倍数下信号的变化来研究信号的动态特性。

（4）小波变换巧妙地利用了非均匀的分辨率，较好地解决了时间和频率分辨率的矛盾；在低频段用高的频率分辨率和低的时间分辨率（宽的分析窗口），而在高频段则用低的频率分辨率和高的时间分辨率（窄的分析窗口），这与时变信号的特征一致。

（5）小波变换将信号分解为在对数坐标中具有相同大小频带的集合，这种以非线性的对数方法而不是以线性方法处理频率的方法对时变信号具有明显的优越性。

（6）小波变换是稳定的，是信号的冗余表示。由于 a、b 是连续变化的，相邻分析窗的绝大部分是相互重叠的，相关性很强。

（7）小波变换与傅里叶变换一样，具有统一性和相似性，其正反变换具有完美的对称性，小波变换具有基于卷积和 QMF 的塔形快速算法。

小波变换具有许多重要而又有应用价值的性质，其中线性特性、平移和伸缩特性、微分运算特性等三个常用性质如下：

1）线性特性

设 $f_1(t)$ 的小波变换为 $W_{f_1}(a,b)$，$f_2(t)$ 的小波变换为 $W_{f_2}(a,b)$，对于小波变换则有如下线性关系。若 $f(t)=k_1 f_1(t)+k_2 f_2(t)$，则有

$$W_f(a,b) = k_1 W_{f_1}(a,b) + k_2 W_{f_2}(a,b) \tag{3-122}$$

2）平移和伸缩特性

设 $f(t)$ 的小波变换为 $W_f(a,b)$，即

$$f(t) \Leftrightarrow W_f(a,b)$$

则有

（1）平移特性：

$$f(t-b_0) \Leftrightarrow W_f(a,b-b_0) \tag{3-123}$$

（2）伸缩特性：

$$f(a_0 t) \Leftrightarrow \frac{1}{\sqrt{a_0}} W_f(a_0 a, a_0 b) \tag{3-124}$$

3）微分运算特性

设 $f(t)$ 的小波变换为 $W_f(a,b)$，则有如下微分性质：

$$W_{\psi_{a,b}}\left(\frac{\partial^n f(t)}{\partial t^n}\right) = (-1)^n \int_{-\infty}^{+\infty} f(t)\left(\frac{\partial^n}{\partial t^n}\right)[\overline{\psi_{a,b}(t)}]\mathrm{d}t \tag{3-125}$$

3.6.3 离散小波变换

1. 一维离散小波变换

连续小波变换通常用于理论分析，在离散的数值计算中，需要对小波变换的尺度因子、位移因子进行离散化。

即取 $a=a_0^m$，$b=nb_0a_0^m$，m，n 为整数，$a_0>0$，$b_0>0$，通常取 $a_0=2$，则一维离散小波定义为

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}}\psi\left(\frac{t-nb_0a_0^m}{a_0^m}\right) = a_0^{-\frac{m}{2}}\psi(a_0^{-m}t-nb_0) \tag{3-126}$$

由此可得一维离散小波变换（DWT）为

$$< f,\psi_{m,n} > = a_0^{-\frac{m}{2}}\int_{-\infty}^{+\infty} f(t)\,\overline{\psi_{m,n}(t)}\mathrm{d}t = a_0^{-\frac{m}{2}}\int_{-\infty}^{+\infty} f(t)\,\overline{\psi(a_0^{-m}t-nb_0)}\mathrm{d}t \tag{3-127}$$

2. 二维离散小波变换

一维离散小波变换可以扩展到二维，这里仅考虑尺度函数是可分离的，在二维情况下，需要一个二维可分离的尺度函数 $\varphi(x,y)$ 和三个可分离的方向敏感的二维小波 $\phi^1(x,y)$、$\phi^2(x,y)$ 和 $\phi^3(x,y)$。分别是一维尺度函数 φ 和 ϕ 相应的小波函数的乘积，即

$$\varphi(x,y) = \varphi(x)\varphi(y) \tag{3-128}$$

$$\phi^1(x,y) = \phi(x)\varphi(y) \tag{3-129}$$

$$\phi^2(x,y) = \varphi(x)\phi(y) \tag{3-130}$$

$$\phi^3(x,y) = \phi(x)\phi(y) \tag{3-131}$$

先分别给定可分离的二维尺度函数及二维小波函数，定义尺度基函数和平移基函数为

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\varphi(2^j x-m, 2^j y-n) \tag{3-132}$$

$$\phi_{j,m,n}^i(x,y) = 2^{j/2}\phi^i(2^j x-m, 2^j y-n) \quad i=1,2,3 \tag{3-133}$$

式中，i，j，m，n 均为正整数，由此可得大小为 $M\times N$ 的函数 $f(x,y)$ 的二维离散小波变换为

$$W_\varphi(j_0,m,n) = \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\varphi_{j_0,m,n}(x,y) \tag{3-134}$$

$$W_{\phi}^{i}(j_0,m,n)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\phi_{j_0,m,n}^{i}(x,y)\quad i=1,2,3 \tag{3-135}$$

式中，j_0 是任意的初始尺度，系数 $W_{\varphi}(j_0,m,n)$ 定义了在尺度 j_0 的 $f(x,y)$ 的近似值，系数 $W_{\phi}^{i}(j_0,m,n)$ 对 $j\geqslant j_0$，附加了三个方向的细节。一般情况令 $j\geqslant j_0$，$M=N=2^J(j=0,1,2,3,\cdots,J-1$ 和 $m,n=0,1,2,3,\cdots,2^j-1)$。

二维离散小波逆变换为

$$f(x,y)=\frac{1}{\sqrt{MN}}\sum_{m}\sum_{n}W_{\varphi}(j_0,m,n)\varphi_{j_0,m,n} + \frac{1}{\sqrt{MN}}\sum_{i=1}^{3}\sum_{j=j_0}^{\infty}\sum_{m}\sum_{n}W_{\phi}^{i}(j_0,m,n)\varphi_{j_0,m,n}(x,y) \tag{3-136}$$

习题

1. 什么是数字图像的变换域处理方法？它的基本思路是什么？

2. 根据傅里叶变换的定义，傅里叶变换存在的条件有哪些？

3. 什么是高斯函数？高斯函数的傅里叶变换具有哪些特点？

4. 什么是功率谱？功率谱的物理意义是什么？

5. 试述什么是傅里叶变换的旋转不变性和可分离性。

6. 选择一幅图像，编写 MATLAB 程序实现以下功能：

(1) 实现图像的二维傅里叶变换，绘出幅值谱和相位谱图像。

(2) 结合理论分析，观察傅里叶变换的频谱分布特点。

(3) 进行傅里叶逆变换重建图像。

(4) 假定相位等于 0，依据幅值谱重构图像。

(5) 假定幅值等于常数，依据相位谱重构图像。

(6) 总结图像傅里叶变换的特点。

7. DCT 变换的主要优点有哪些？

8. 编写 MATLAB 程序对一幅图像进行 DCT 变换，分析其频谱分布特点。

9. 什么是 K-L 变换？为什么说 K-L 变换是最佳变换？

10. K-L 变换消除图像数据相关性的原理是什么？

11. 简述 K-L 变换如何控制重构图像的误差。

12. 什么是沃尔什变换？沃尔什变换具有哪些主要特点？

13. 什么是哈达玛变换？它与沃尔什变换有何区别和联系？

14. 不同阶数的哈达玛变换矩阵有何关系？

15. 什么是小波？小波具有哪些特性？

16. 试简述小波变换的主要特点。

17. 简要说明由连续小波变换推出离散小波变换时，如何对一维小波变换的尺度因子、位移因子进行离散化。

18. 应用 MATLAB 对一幅图像进行小波变换，分析和总结图像小波变换的特点。

第4章 图像处理的基本运算

CHAPTER 4

图像处理是建立在各种算法基础上的处理方法，这些算法无论简单还是复杂，常常涉及图像像素的灰度值运算和几何位置关系运算。基于像素的灰度值以及几何位置关系运算都属于图像处理中的基本运算。本章将围绕数字图像处理的基本运算，主要介绍图像处理中的点运算、代数运算和几何变换(齐次坐标体系、图像平移、旋转变换、比例缩放、镜像变换、图像复合变换和透视变换等)及其在图像处理中的应用等。

4.1 基本运算的类型

在数字图像处理中，经常需要采用各种各样的算法。根据数字图像处理运算中输入信息与输出信息的类型，具有代表性的图像处理典型算法从功能上包括以下几种：

(1) 单幅图像⟶单幅图像。

(2) 多幅图像⟶单幅图像。

(3) 单幅或多幅图像⟶数值/符号等。

在以上三类运算形式中，所有输入信息都是图像且其灰度值都是非负整数值，而输出信息的形式则各不相同，既可以是具有非负灰度值的数字图像，又可以是仅具有0、1两个灰度值的二值图像，也可以是对输入图像逐个像素点进行解释的符号或由特定参数组成的某种二维信息(又称为标号图像)，还可以是从图像中提取出的以数值或符号描述的特征信息。所有以二维信息形式输出的信息统称为广义图像，标号图像也属于广义图像的范畴。

三类运算中，第一类运算功能是数字图像处理技术中最基本的功能。对基本的图像处理功能，根据输入图像得到输出图像(目标图像)处理运算的数学特征，可将图像处理运算方式分为点运算、代数运算和几何运算。这些运算都是基于空间域的图像处理运算，与空间域运算相对应的是变换域运算，将在后续章节中讨论。

4.2 点运算

在图像处理中，点运算(point operation)是一类简单却非常具有代表性的重要算法之一，也是其他图像处理运算的基础。运用点运算可以改变图像数据所占据的灰度值范围。对于一幅输入图像，经过点运算会产生一幅输出图像，输出图像中每个像素点的灰度值仅由

相应输入像素点的灰度值确定。这与邻域处理算法截然不同，在邻域算法中，每个输出像素的灰度值由对应输入像素的一个邻域内若干像素点的灰度值共同决定。因此，点运算不会改变图像内的空间位置关系。点运算是图像数字化软件以及图像显示软件的重要组成部分。

4.2.1　点运算的种类

点运算从数学上可以分为线性点运算和非线性点运算两类。

1. 线性点运算

线性点运算指输入图像的灰度级与目标图像的灰度级呈线性关系。线性点运算的灰度变换函数形式可以采用线性方程描述，即

$$D_B = aD_A + b \tag{4-1}$$

式中，D_A 为输入点的灰度值；D_B 为相应输出点的灰度值。显然，这种线性运算关系可用图 4-1 表示。

(1) 如果 $a=1, b=0$，则只需将输入图像复制到输出图像即可。若 $a=1$，而 $b\neq0$，则操作仅使所有像素的灰度值上移或下移，其效果是使整个图像在显示时更暗或更亮。

(2) 如果 $a>1$，则输出图像的对比度增大。

(3) 如果 $a<1$，则输出图像的对比度减小。

(4) 如果 $a<0$，即 a 为负值，则暗区域变亮，亮区域变暗，点运算完成图像求补。

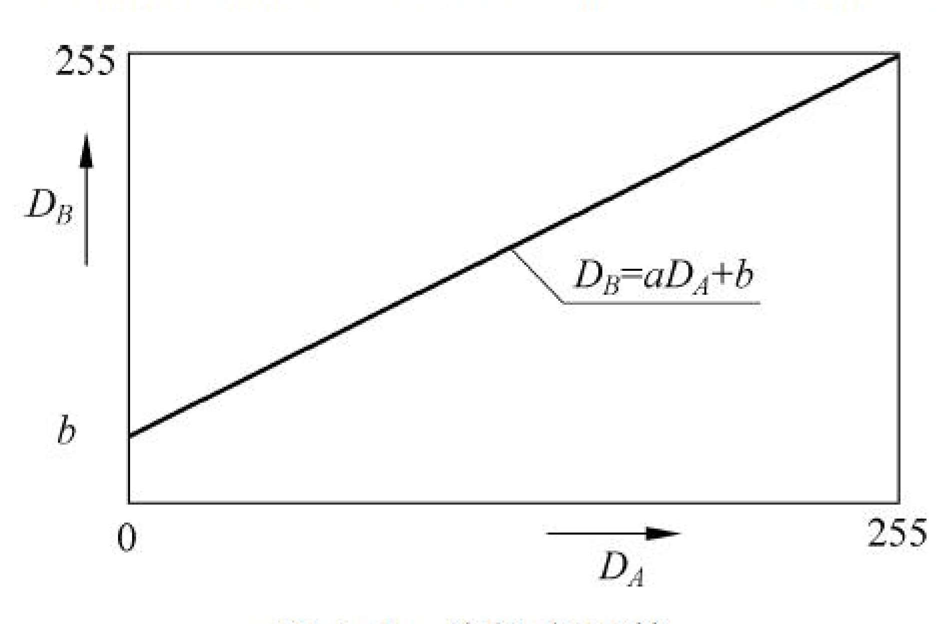


图 4-1　线性点运算

2. 非线性点运算

除了线性点运算外，还有非线性点运算。一般考虑非减(nondecreasing)的灰度变换函数，其灰度变换关系如图 4-2 所示。非线性点运算灰度变换函数的斜率处处为正数，这类函数保留了图像的基本外貌。

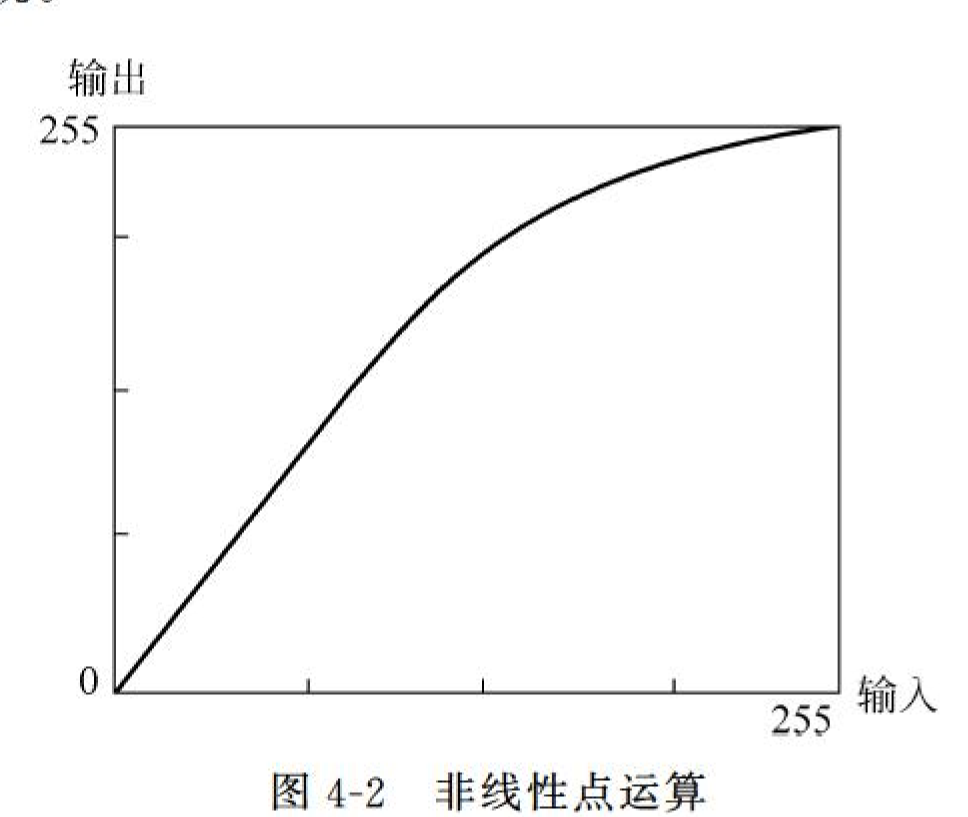


图 4-2　非线性点运算

非线性点运算的函数形式可以表示为

$$D_B = f(D_A) \tag{4-2}$$

式中，D_A 为输入点的灰度值；D_B 为相应输出点的灰度值；f 表示非线性函数，函数表达式需根据具体应用选择有代表性的非线性函数形式。式(4-3)～式(4-5)是三种典型的非线性点运算函数。

$$D_B = f(D_A) = D_A + CD_A(D_m - D_A) \tag{4-3}$$

式中，D_m 为灰度级的最大值；参数 C 定义了中间灰度范围内的增加量($C>0$)或减少量($C<0$)。非线性点运算可增加中间范围像素的灰度级而只使暗像素和亮像素进行较小改变。

$$f(D_A) = \frac{D_m}{2}\left\{1 + \frac{1}{\sin\left(\frac{\alpha\pi}{2}\right)}\sin\left[\alpha\pi\left(\frac{D_A}{D_m} - \frac{1}{2}\right)\right]\right\}, \quad 0<\alpha<1 \tag{4-4}$$

其中，灰度级范围为 $0\sim D_m$，在该灰度级范围内，直方图非零。参数 α 越大，效果越明显。该函数是基于正弦函数的非线性点运算形式。这类非线性单调点运算通过降低较亮或较暗物体的对比度来加强灰度级处于中间范围的物体的对比度，这类函数曲线为 s 形，其灰度变换函数在中间部分的斜率大于 1，而两端处斜率小于 1。

$$f(D_A) = \frac{D_m}{2}\left\{1 + \frac{1}{\tan\left(\frac{\alpha\pi}{2}\right)}\tan\left[\alpha\pi\left(\frac{D_A}{D_m} - \frac{1}{2}\right)\right]\right\}, \quad 0<\alpha<1 \tag{4-5}$$

该函数是基于正切函数的非线性点运算表达式，同样，参数 α 决定点运算的效果。这类非线性单调点运算是通过压低在中间灰度级区间的对比度，而将较亮和较暗部分的对比度加强。该灰度变换函数在中间处的斜率小于 1，而在靠近两端处斜率大于 1。

4.2.2 点运算与直方图

点运算是一种在确定的函数关系下所进行的像素变换运算，因此，点运算之后输出图像和输入图像之间的直方图也具有与变换函数相关联的对应关系。

设点运算之间的函数关系为 $D_B=f(D_A)$，输入直方图、灰度变换函数以及输出直方图之间的关系如图 4-3 所示，灰度值 D_A 转换为 D_B，同样，灰度值 $D_A+\Delta D_A$ 转换为 $D_B+\Delta D_B$，且在灰度级$[D_A, D_A+\Delta D_A]$之间的所有像素被转化到灰度级$[D_B, D_B+\Delta D_B]$之间，进一步可得出，灰度级$[D_A, D_A+\Delta D_A]$之间的输入像素的个数等于灰度级$[D_B, D_B+\Delta D_B]$之间输出像素的个数，即

$$\int_{D_A}^{D_A+\Delta D_A} H_A(D)\mathrm{d}D = \int_{D_B}^{D_B+\Delta D_B} H_B(D)\mathrm{d}D$$

当 ΔD_A 很小时，ΔD_B 的取值也较小，因此，输入、输出图像阴影部分面积可分别用两个小矩形的面积近似替代。

$$H_B(D_B)\Delta D_B = H_A(D_A)\Delta D_A \tag{4-6}$$

因此，输出直方图的值为

$$H_B(D_B) = H_A(D_A)\Delta D_A/\Delta D_B = \frac{H_A(D_A)}{\Delta D_B/\Delta D_A}$$

当 ΔD_A 趋于 0 时，取极限，因而可用微分表示为

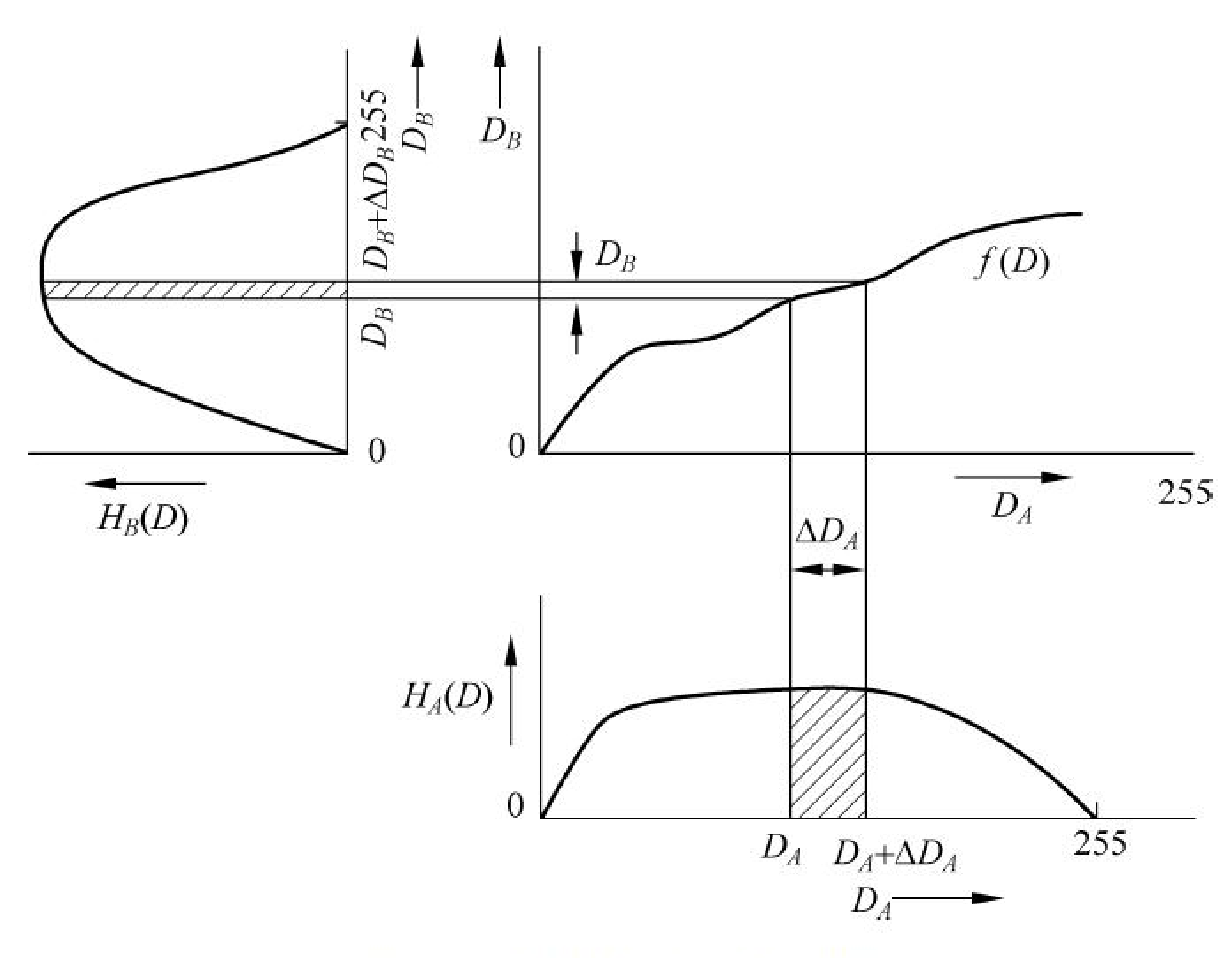


图 4-3 点运算与直方图之关系

$$H_B(D_B)=\frac{H_A(D_A)}{\frac{\mathrm{d}D_B}{\mathrm{d}D_A}}$$

代入点运算的函数形式得

$$H_B(D_B)=\frac{H_A(D_A)}{\frac{\mathrm{d}f(D_A)}{\mathrm{d}D_A}} \tag{4-7}$$

4.2.3 点运算的应用

由于点运算能有规律地改变像素点的灰度值，因而点运算有时又被称为对比度增强或灰度变换(gray-scale transformation，GST)。因此，通过恰当定义数学运算的形式，点运算可用于改善图像数字化设备或图像数字显示设备的某些局限性。

1. 对比度增强

在一些数字图像中，技术人员所关注的特征可能仅占据整个灰度级范围非常小的一部分。点运算可以扩展所关注部分的灰度信息的对比度，使之占据可显示灰度级的更大部分。该方法有时称为对比度增强(contrast enhancement)或对比度拉伸(contrast stretching)。

2. 光度学标定

人们常希望数字图像的灰度能反映诸如光照强度、光密度等某些物理特性，通过去掉图像传感器的非线性的影响，点运算可达到该目的。例如，假设一幅图像被一个对光照强度呈非线性关系的仪器所数字化，点运算可以通过适当的灰度变换运算，使灰度级与光照强度的等步长增量匹配。

点运算的另一个用处是变换灰度的单位。假定有一个图像数字化器，用来数字化一幅显微镜下观察到的图像。其产生的灰度值与标本的透射率呈线性关系，点运算可用来产生一幅图像，该图像的灰度级可代表光学密度的等步长增量。光度学标定通常作为图像数字化的软件部分。

3. 显示标定

一些显示设备通常具有能突出图像视觉特征的优选灰度范围。使用这样的显示设备时，数字图像中具有相同对比度的较暗和较亮的特征，在显示时却不能以同样的性能表现出来。在这种情况下，用户可利用点运算让感兴趣的所有特征同等突出地显示出来。

一些显示设备不能保持数字图像上像素的灰度值和显示屏幕上相应点的亮度之间的线性关系，同样，许多胶片记录仪不能线性地将灰度值转换为光密度。这一缺点也可以通过点运算予以克服，即在图像显示之前，先设计合理的点运算关系。另外，可将点运算和显示非线性组合起来互相抵消，以保持在显示图像时的线性关系。这一过程就是所谓显示标定(display calibration)。

少数情况下，非线性显示关系对于图像的表示也具有一定的作用。例如，电视机或CRT 显示器的 γ 校正就是利用了这种非线性关系，点运算可纠正或调整显示的 γ 值。

点运算有时被视为强化细节或增加图像某些部分的对比度的图像处理步骤。然而，由于信息实际上包含在数字图像中，所以，实际要做的工作是使感兴趣部分的灰度级与显示设备的对比度范围匹配起来。显示标定和对比度增强也经常作为数字图像显示的软件部分。

4. 轮廓线

点运算可为图像加上轮廓线。可以应用点运算进行阈值化，根据灰度级可将一幅图像划分成一些不连接的区域，有助于在后续处理中确定边界或用于定义掩膜。

5. 剪裁

因为数字图像通常以整型格式存储，所以，可用的灰度级范围是有限的。对于 8bit 图像，在每个像素值被存储之前，输出灰度级一定要被裁剪到 0～255 的范围之内。

4.3 代数运算

代数运算是指对两幅输入图像进行点对点的加、减、乘、除运算而得到目标图像的运算。另外，还可通过适当的组合，形成涉及几幅图像的复合代数运算方程。图像处理代数运算的四种基本形式分别如下：

$$C(x,y) = A(x,y) + B(x,y) \tag{4-8}$$

$$C(x,y) = A(x,y) - B(x,y) \tag{4-9}$$

$$C(x,y) = A(x,y) \times B(x,y) \tag{4-10}$$

$$C(x,y) = A(x,y) \div B(x,y) = \frac{A(x,y)}{B(x,y)} \tag{4-11}$$

式中，$A(x,y)$和 $B(x,y)$为输入图像表达式；$C(x,y)$为输出图像表达式。某些情况下，输入图像之一也可以是常数。在一些特定情况下，参与代数运算的输入图像可能多于两个，如用于消除加性随机噪声的图像相加运算一般都多于两个输入图像。

4.3.1 代数运算的意义

在数字图像处理技术中，代数运算具有非常广泛的应用和重要的意义。例如图像相加运算的重要应用之一是对同一场景的多幅图像求平均值的运算，它可以被用于消除或降低加性随机噪声，并具有良好的效果。图像相加也可用于将一幅图像的内容叠加到另一幅图

像上，从而实现二次曝光(double-exposure)。

图像相减运算可用于消除一幅图像中所不需要的加性图像，加性图像可能是缓慢变化的背景阴影、周期性噪声，或在图像上每一像素点均已知的附加污染等。减法运算还可用于检测同一场景中的两幅图像之间的变化。例如，通过对同一场景的序列图像的相减运算可用于检测物体的运动。在计算用于确定物体边界位置的梯度时，也需要应用图像的相减运算。

在数字图像处理中，虽然乘、除运算应用得相对少一些，但它们也具有很重要的应用。例如，在获取数字化图像过程中，图像数字化设备对一幅图像各点的敏感程度不可能完全相同，乘、除运算可用于纠正这方面的不利影响。此外，乘法运算在获取图像的局部图案时发挥作用，用一幅掩膜图像(mask image)乘以某一图像可遮住该图像中的某些部分，使其仅保留图像中感兴趣的部分，而除法运算还可以产生对颜色和多光谱图像分析十分重要的比率图像。

4.3.2 加法运算

加法运算通常用于平均值降噪等多种场合。图像相加一般用于对同一场景的多幅图像求平均，以便有效地降低加性噪声，通常，图像采集系统在采集图像时有这类参数可供选择。对于一些经过长距离模拟通信方式传送的图像(如太空航天器传回的星际图像)这种处理是不可缺少的。当噪声可以用同一个独立分布的随机模型表示和描述时，则利用求平均值方法降低噪声信号、提高信噪比非常有效。

在实际应用中，要得到一静止场景或物体的多幅图像是比较容易的。如果这些图像被一加性随机噪声源所污染，则可通过对多幅静止图像求平均值来达到消除或降低噪声的目的。在求平均值的过程中，图像的静止部分不会改变，而由于图像的噪声是随机性的，各不相同的噪声图案累积得很慢，因此可以通过多幅图像求平均值降低随机噪声的影响。

若有一静止场景的图像被加性随机噪声污染，且已获得由 k 幅该静止场景图像组成的图像集合，设图像可表示为

$$D_i(x,y) = S(x,y) + N_i(x,y) \tag{4-12}$$

式中，$S(x,y)$为静止场景的理想图像；$N_i(x,y)$表示由于胶片的颗粒或数字化系统中的电子噪声所引起的噪声图像。集合中的每幅图像被不同的噪声图像所污染。虽然并不能准确获取这些噪声信息，但通常情况下图像的噪声信号都来自于同一个互不相干且均值等于零的随机噪声样本集。设 $P(x,y)$表示功率信噪比，对于图像中的任意像素点，则 $P(x,y)$可定义为

$$P(x,y) = \frac{S^2(x,y)}{E[N^2(x,y)]} \tag{4-13}$$

对 K 幅图像求平均，可得

$$\overline{D}(x,y) = \frac{1}{K}\sum_{i=1}^{K}[S(x,y) + N_i(x,y)] \tag{4-14}$$

平均值图像的功率信噪比为

$$\overline{P}(x,y) = \frac{S^2(x,y)}{E\left\{\left[\frac{1}{K}\sum_{i=1}^{K}N_i(x,y)\right]^2\right\}} \tag{4-15}$$

$N_i(x,y)$为随机噪声，因而具有以下特性：

$$E[N_i(x,y)] = 0 \tag{4-16}$$

$$E[N_i(x,y) + N_j(x,y)] = E[N_i(x,y)] + E[N_j(x,y)] \tag{4-17}$$

$$E[N_i(x,y)N_j(x,y)] = E[N_i(x,y)]E[N_j(x,y)] \tag{4-18}$$

由此，可以得出

$$\bar{P}(x,y) = \frac{S^2(x,y)}{\frac{1}{K^2}KE[N^2(x,y)]} = KP(x,y) \tag{4-19}$$

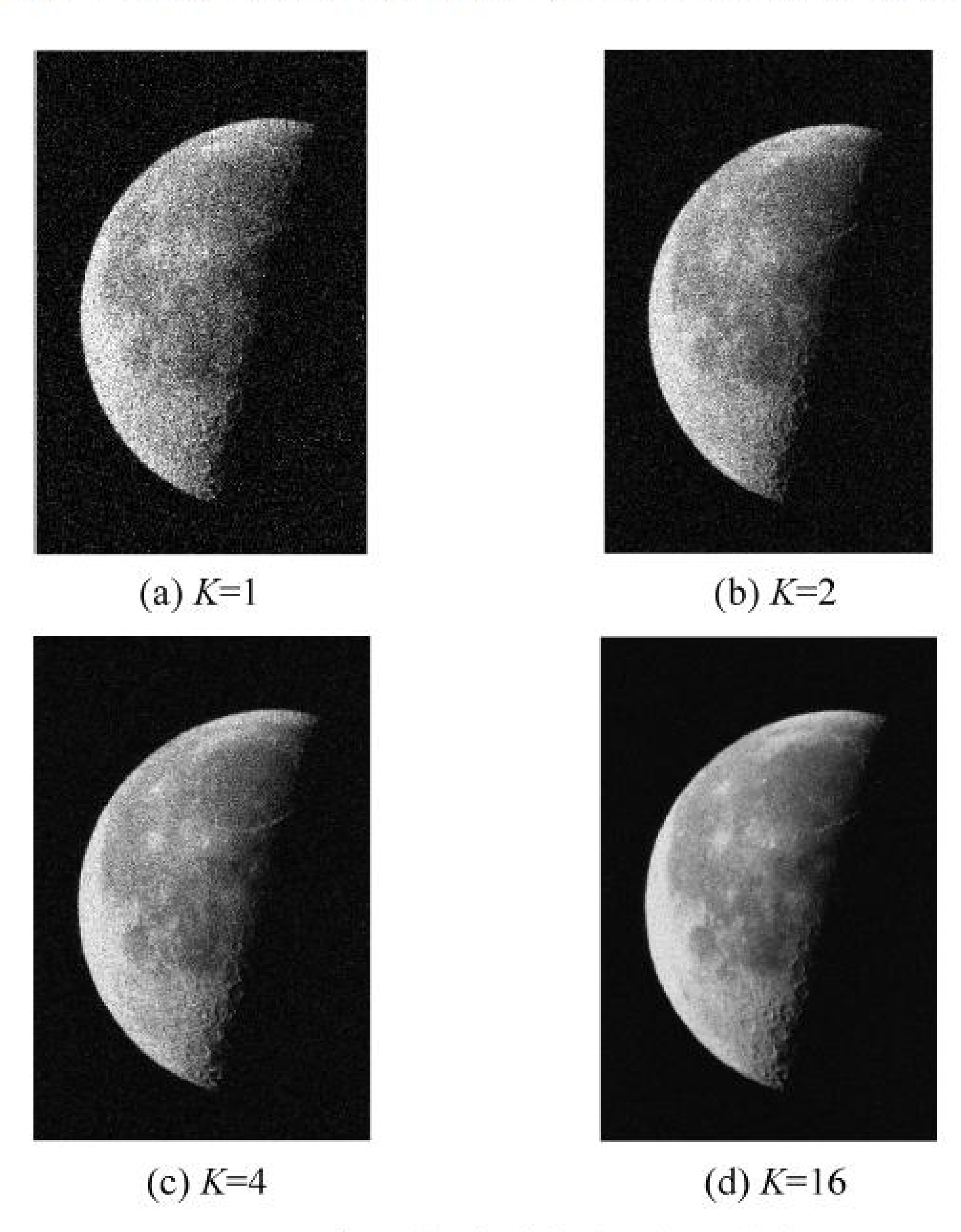
因此，对 K 幅图像进行平均，则图像中每一点的功率信噪比提高了 K 倍。而功率信噪比与幅度信噪比（SNR）之间是平方关系。故有

$$\mathrm{SNR} = \sqrt{\bar{P}(x,y)} = \sqrt{K} \cdot \sqrt{P(x,y)} \tag{4-20}$$

即求平均值以后，图像的幅值信噪比单幅图像提高了$\sqrt{K}$倍，幅度信噪比随着被平均图像数量的增加而提高。

例 4-1：图像加法消除噪声的示例

如图 4-4 所示是加法消除噪声的示例。在图 4-4(a)中 $K=1$，表示 1 幅噪声图像平均，即具有加性噪声的原始月球照片；图 4-4(b)中 $K=2$，表示 2 张照片相加后求平均；图 4-4(c)中 $K=4$，表示 4 张照片相加后求平均；图 4-4(d)中 $K=16$，表示 16 张照片相加后求平均。由于相加图像越来越多，SNR 值不断提高，因此，图像质量由(a)到(d)明显提高。



(a) K=1　(b) K=2　(c) K=4　(d) K=16

图 4-4　求平均消除加性随机噪声

4.3.3 减法运算

图像相减常用于检测变化及运动物体，图像相减运算又称为图像差分运算。差分方法可以分为可控制环境下的简单差分方法和基于背景模型的差分方法。在可控制环境下，或

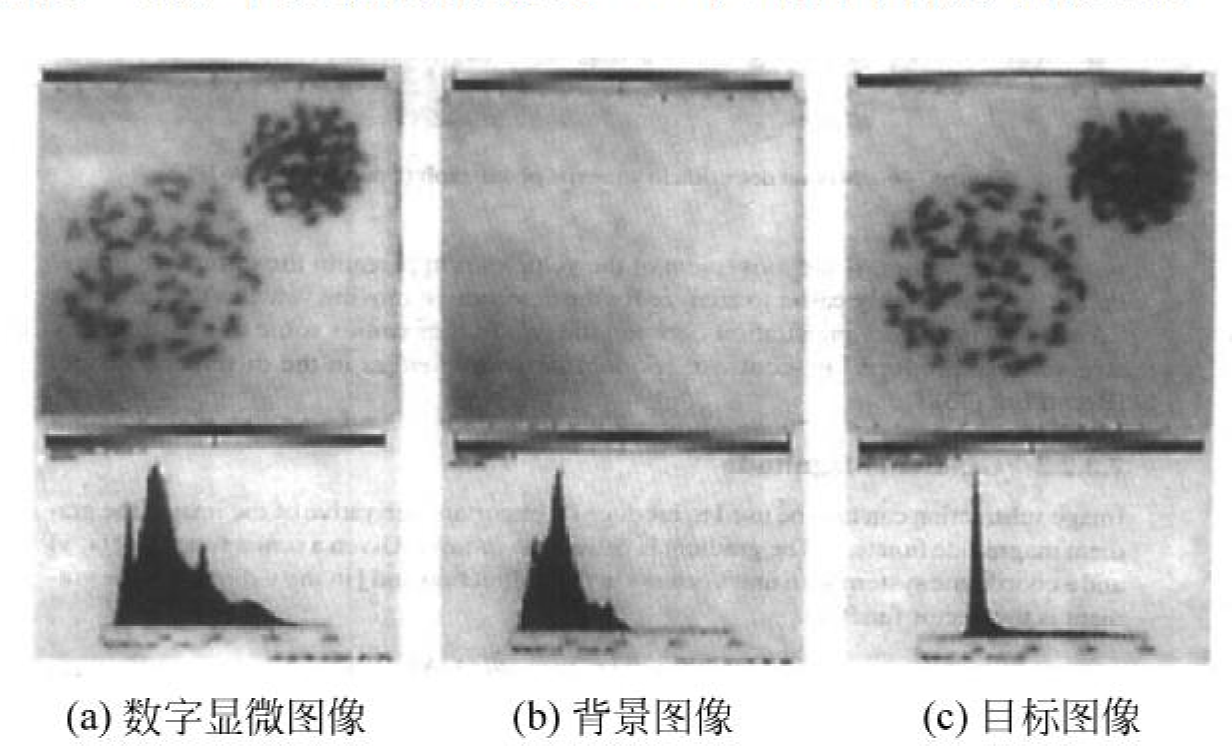
者在很短的时间间隔内，可以认为背景是固定不变的，可以直接使用差分运算检测变化及运动物体。

1. 消除背景影响

在有些情况下，背景对图像中的被研究物体具有不利影响，这时背景就成为了噪声，可以利用图像减法运算消除背景噪声的影响。

例 4-2：减法消除背景噪声

如图 4-5 所示，图 4-5(a)表示一幅数字化的光学显微图像(在每幅图像的下面为其各自的灰度直方图)，该图像含有两群人的染色体，图像被缓慢变化的背景影调图像所污染；在图 4-5(b)中，显微镜载物台被移出使得物镜下形成空白区域，因此，图像(b)只含有背景影调图案；图像(c)是从图(a)中被减去背景图像(b)，对每个像素点的灰度值再加上一个常数，使图像的整体亮度一致，本实验的常数为 64 时，结果图像具有大致一致的亮度。



(a) 数字显微图像　　(b) 背景图像　　(c) 目标图像

图 4-5　消除背景图像的应用

注意图 4-5(c)中的直方图与理想的直方图略有不同，特别是一些像素的灰度值小于理论上的最小值 64。这是由于数字化过程中的噪声所引起的，数字化器的噪声使背景在图 4-5(a)与(b)中并不具有完全相同的灰度值。

2. 差影法

所谓差影法，实际上就是图像的减法运算(又称减影技术)，是指将同一景物在不同时间拍摄的图像或同一景物在不同波段的图像相减。差值图像提供了图像间的差异信息，能用于指导动态监测、运动目标的检测和跟踪、图像背景的消除及目标识别等。

差影法在自动现场监测等领域具有广泛的运用。可以应用在监控系统中，在银行金库内，摄像头每间隔某一固定时间(如 10s)拍摄一幅图像，并与上一幅图像进行差影运算，如果图像差别超过了预先设置的阈值，则表明可能有异常情况发生，应自动或以某种方式报警。差影法可用于检测变化目标及遥感图像的动态监测；利用差值图像可以发现森林火灾、洪水泛滥，监测灾情变化及估计损失等；也可用于监测河口、海岸的泥沙淤积及监视江河、湖泊、海岸等的污染。利用差值图像还能鉴别出耕地及不同的作物覆盖情况。

差影技术还可用于消除图像背景。例如，该技术可用于诊断印刷线路板及集成电路掩膜的缺陷。特别是用于血管造影技术中，肾动脉造影术对诊断肾脏疾病有独特优势。为了减少误诊，人们希望提供反映游离血管的清晰图像。通常的肾动脉造影在造影剂注入后，虽然能够看出肾动脉血管的形状及分布，但由于肾脏周围血管受到脊椎及其他组织影像的重

叠，难以得到理想的游离血管图像。对此，可摄制出肾动脉造影前后的两幅图像，相减后就能把脊椎及其他组织的影像去掉，而仅保留血管图像。此外，电影特技中应用的"蓝幕"技术，其实也包含差影法的基本原理。

在某些特定情况下，差影法可以用于混合图像的分离，如例 4-3 所示。

例 4-3：差影法分离图像

如图 4-6 所示是图像减法运算的示例。图 4-6(a)为某种原因所形成的混合图像，已知该混合图像是由(b)和(c)叠加而成(或由于两次曝光而成)，因此，可以利用差影法，以图 4-6(a)和图 4-6(b)进行差影，即可得到图 4-6(c)所示结果。



(a) 混合图像


(b) 被减图像


(c) 差影图像

图 4-6　差影法用于图像分离

图像在进行差影法运算时必须使两相减图像的对应点位于空间同一目标点上。否则，必须先作几何校正与配准再进行求差运算。当将一个场景中的系列图像相减用来检测运动或其他变化时，难以保证准确对准，这时就需要更进一步的分析。例如，设差图像由下式给定：

$$C(x,y)=A(x,y)-A(x+\Delta x,y) \tag{4-21}$$

如果 Δx 很小，那么式(4-21)可以近似为

$$C(x,y)\approx\frac{\partial}{\partial x}A(x,y)\Delta x \tag{4-22}$$

由于$\frac{\partial}{\partial x}A(x,y)$本身也是一幅图像，其直方图以 $H'(D)$表示，因此，式(4-22)表示的位移差图像的直方图为

$$H_c(D)\approx\frac{1}{\Delta x}H'_A\left(\frac{D}{\Delta x}\right) \tag{4-23}$$

式(4-23)表明，减去稍微有些对不准的原图像的复制图像，可以得到偏导数图像，偏导数的方向为图像位移方向。

3. 求梯度幅度

图像的减法运算也可应用于求图像梯度函数。梯度是数学与场论中的概念，它是向量函数，前述章节中已经介绍了关于梯度的概念，其定义形式为

$$\nabla f(x,y)=\mathrm{i}\frac{\partial f}{\partial x}+\mathrm{j}\frac{\partial f}{\partial y} \tag{4-24}$$

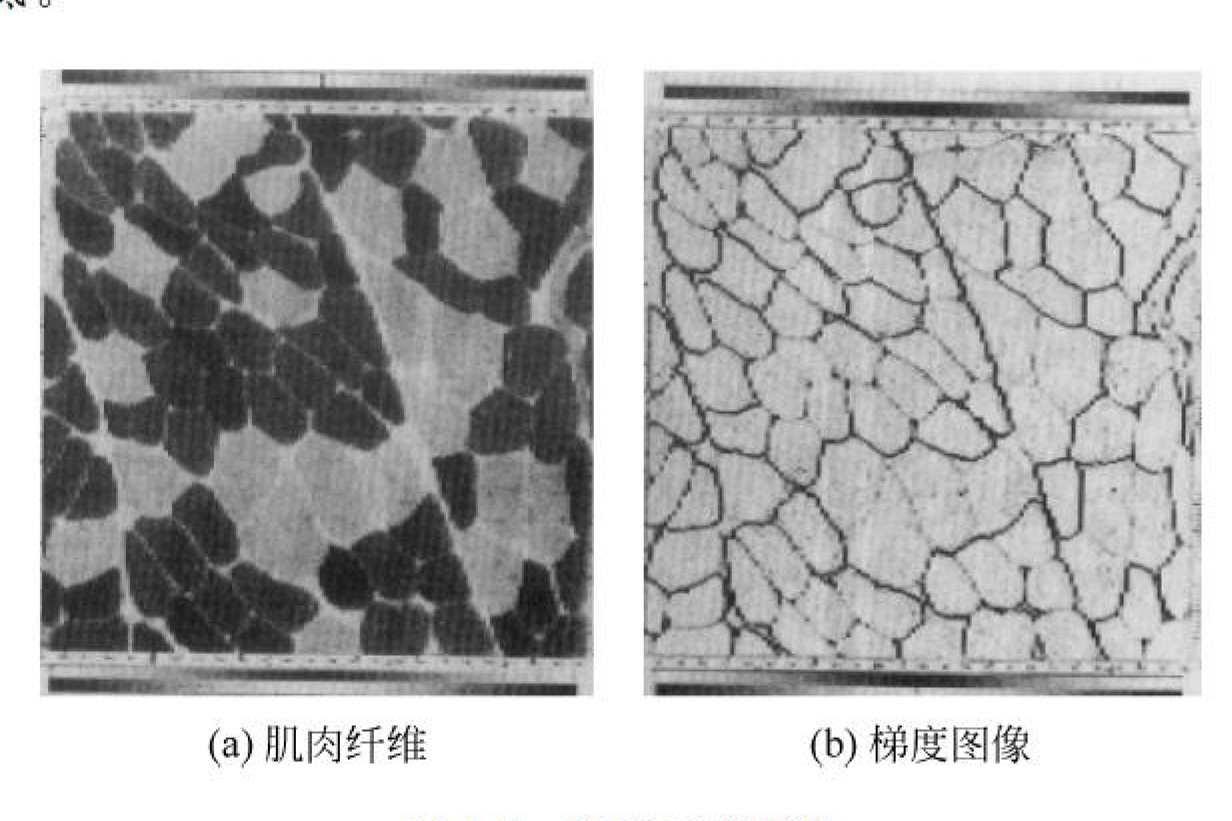
梯度幅度可由下式表示：

$$|\nabla f(x,y)|=\sqrt{\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2} \tag{4-25}$$

考虑到运算的方便性，梯度幅度可由下式近似计算：

$$|\nabla f(x,y)|=\max[|f(x,y)-f(x+1,y)|-|f(x,y)-f(x,y+1)|] \tag{4-26}$$

也就是说，梯度可近似取值为水平方向相邻像素之差的绝对值和垂直方向相邻像素之差的绝对值中的较大者。在图像中物体的边缘属于斜率陡峭之处，梯度值一般较大。梯度运算可以获取图像内部的边缘信息。如图4-7所示是图像梯度运算的示例。对于肌肉活组织检查标本的细胞显微图像之梯度幅度图，梯度幅度在边缘处很高，而在均匀的肌肉纤维的内部，梯度幅度很低。



(a) 肌肉纤维　　(b) 梯度图像

图4-7　梯度幅度图像

4.3.4 乘法运算

乘法运算可用来遮住图像的某些部分，其典型运用是用于获得掩膜图像。对于需要保留下来的区域，掩膜(mask)图像的值置为1，而在需要被抑制掉的区域，掩膜图像的值置为0。乘法获取掩膜图像如例4-4所示。

例4-4：乘法获取掩膜图像

如图4-8所示是图像乘法运算的示例。原图像乘上掩膜图像，可抹去图像的某些部分，即使该部分为0。然后可利用一个互补的掩膜来抹去第二幅图像中的另一些区域，而这些区域在第一幅图像中被完整地保留了下来。两幅经过掩膜的图像相加可得最终结果。图4-8(c)为图4-8(a)与图4-8(b)相乘的结果。

一般情况下，利用计算机图像处理软件生成掩膜图像的步骤如下：

(1) 新建一个与原始图像大小相同的图层，图层文件一般保存为二值图像文件。

(2) 用户在新建图层上人工勾绘出所需要保留的区域，区域的确定也可以由其他二值图像文件导入或由计算机图形文件(矢量)经转换生成。

(3) 确定局部区域后，将整个图层保存为二值图像，选定区域内的像素点值为1，非选定区域像素点值为0。



(a) 原图像



(b) 掩膜图像



(c) 局部图像

图 4-8 乘法运算在掩膜图像中的应用

(4) 将原始图像与(3)形成的二值图像进行乘法运算，即可将原始图像选定区域外像素点的灰度值置 0，而选定区域内像素的灰度值保持不变，得到与原始图像分离的局部图像，即掩膜图像。

掩膜技术也可以灵活应用，如可以增强选定区域外的图像而对区域内的图像不做处理，这时，只需将二值图像中区域外像素点置 1 而区域内的像素点置 0 即可。

掩膜图像技术还可以应用于图像局部增强，一般的图像增强处理都是对整幅图像进行操作，但在实际应用中，往往需要只对图像的某一局部区域进行增强，以突出某一具体的目标，若这些局部区域所包含的像素点数目相对于整幅图像来讲非常小，在计算整幅图像的统计量时其影响几乎可以忽略不计，因此以整幅图像的变换或转移函数为基础的增强方法对这些局部区域的影响也非常小，难于达到理想的增强效果。

4.3.5 除法运算

除法运算的典型运用是比值图像处理。例如，除法运算可用于校正成像设备的非线性影响，在特殊形态的图像(如 CT 为代表的医学图像)处理中用到；此外，除法运算还经常用于消除图像数字化设备随空间变化所产生的影响，并可用于产生多光谱图像处理中非常有用的比率图像，有兴趣的读者可参阅这方面的专著。

4.4 几何运算

几何运算又称为几何变换，它是图像处理和图像分析的重要内容之一。通过几何运算，可以根据应用的需要使原图像产生大小、形状和位置等各方面的变化。简单地说，几何变换可改变像素点所在的几何位置，以及图像中各物体之间的空间位置关系，这种运算可以被看

成是将各物体在图像内移动，特别是图像具有一定的规律性时，一个图像可以由另一个图像通过几何变换来产生。几何变换的效果可形象化地理解为把图像看成一张橡皮膜，对该膜做任意伸展，并在若干点把变形后的橡皮膜固定下来的结果。实际上，几何变换的结果可以比上述比喻灵活得多，一个不受约束的几何变换，可将输入图像中的一个点变换到输出图像中任意位置。几何变换不仅提供了产生某些特殊图像的可能，甚至还可以使图像处理程序设计简单化。从变换性质分，几何变换可以分为图像的位置变换（平移、镜像、旋转）、形状变换（放大、缩小和错切）以及图像的复合变换等。

4.4.1 齐次坐标

数字图像是对一幅连续图像的坐标和色彩都离散化了的图像，坐标的离散通过采样完成，色彩的离散通过量化完成，因此可以用二维数组 $f(x,y)$ 表示，其中 x 和 y 表示像素点的坐标位置，$f(x,y)$ 代表图像在点 (x,y) 的灰度值（如果所处理的是彩色图像，则通过 RGB 值表示）。图像几何变换的基础是代数和几何学。几何变换可以使用户获得大小、形状和位置都发生变化的各种图像。为讨论和研究问题的方便，无论图像比例缩放、旋转、反射和错切，还是图像的平移、透视变换及复合变换等，几何变换一般采用如下形式：

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \boldsymbol{T}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{4-27}$$

根据几何学知识，上述变换可以实现图像各像素点以坐标原点的比例缩放、反射、错切和旋转等各种变换，但是上述 2×2 变换矩阵 $\boldsymbol{T}$ 不能实现图像的平移以及绕任意点的比例缩放、反射、错切和旋转等变换。

因此，为了能够用统一的矩阵线性变换形式表示和实现这些常见的图像几何变换，就需要引入一种新的坐标，即齐次坐标。采用齐次坐标可以实现上述各种几何变换的统一表示。

若将点 $A_0(x_0,y_0)$ 在水平方向（x 方向）平移 Δx 距离，在垂直方向（y 方向）平移 Δy 距离到新的位置 $A_1(x_1,y_1)$，如图 4-9 所示，则新位置 $A_1(x_1,y_1)$ 点的坐标为

$$\begin{cases} x_1 = x_0 + \Delta x \\ y_1 = y_0 + \Delta y \end{cases} \tag{4-28}$$

图 4-9 像素点的平移

式(4-28)可以表示为

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{4-29}$$

但却无法表示为式(4-27)的形式，即不能表示为

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

由于矩阵 $\boldsymbol{T}$ 中没有引入平移常量，无论 a、b、c、d 取什么值，都不能实现式(4-28)所示的平移功能。因此，为了用式(4-27)表示平移变换，需要对式(4-27)进行改进。

根据矩阵运算规律，可以将 $\boldsymbol{T}$ 矩阵扩展为如下 2×3 变换矩阵，其形式为

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix} \tag{4-30}$$

若将矩阵 $\boldsymbol{T}$ 分块，该矩阵的第一、二列构成单位矩阵，第三列元素分别为 x、y 方向的平移量。扩展后变换矩阵为 2×3 矩阵，而矩阵相乘时要求前者的列数与后者的行数相等，因此，应在坐标列矩阵 $[x \quad y]^{\mathrm{T}}$ 中引入第三个元素，扩展为 3×1 的列矩阵 $[x \quad y \quad 1]^{\mathrm{T}}$。于是，以 $(x,y,1)$ 表示二维坐标点 (x,y)，就可以实现点的平移变换。变换形式为

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{4-31}$$

通过上述变换虽然可以实现图像各像素点的平移变换，但为变换运算时更方便，一般将 2×3 阶变换矩阵 $\boldsymbol{T}$ 进一步扩充为 3×3 方阵，即采用如下变换矩阵：

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \tag{4-32}$$

于是，平移变换可以用如下形式表示：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{4-33}$$

由此可知，引入附加坐标后，将 2×2 矩阵扩展为 3×3 矩阵，就可以对各种几何变换进行统一表示。这种以 $n+1$ 维向量表示 n 维向量的方法称为齐次坐标表示法。齐次坐标的几何意义相当于点 (x,y) 投影在 xyz 三维立体空间的 $z=1$ 的平面上。

4.4.2 图像平移

平移是日常生活中最普遍的方式之一，例如开学时教室里课桌的重新摆放等都可以视为平移运动。图像的平移是将一幅图像上的所有像素点都按给定的偏移量沿 x 方向(方向水平)和 y 方向(垂直方向)进行移动，如图 4-10 所示。图像的平移是图像几何变换中最简单的变换之一。

若点 $A_0(x_0,y_0)$ 进行平移后，被移到 $A(x,y)$ 位置，其中 x 方向的平移量为 Δx，y 方向的平移量为 Δy，那么，点 $A(x,y)$ 的坐标为

$$\begin{cases} x = x_0 + \Delta x \\ y = y_0 + \Delta y \end{cases}$$

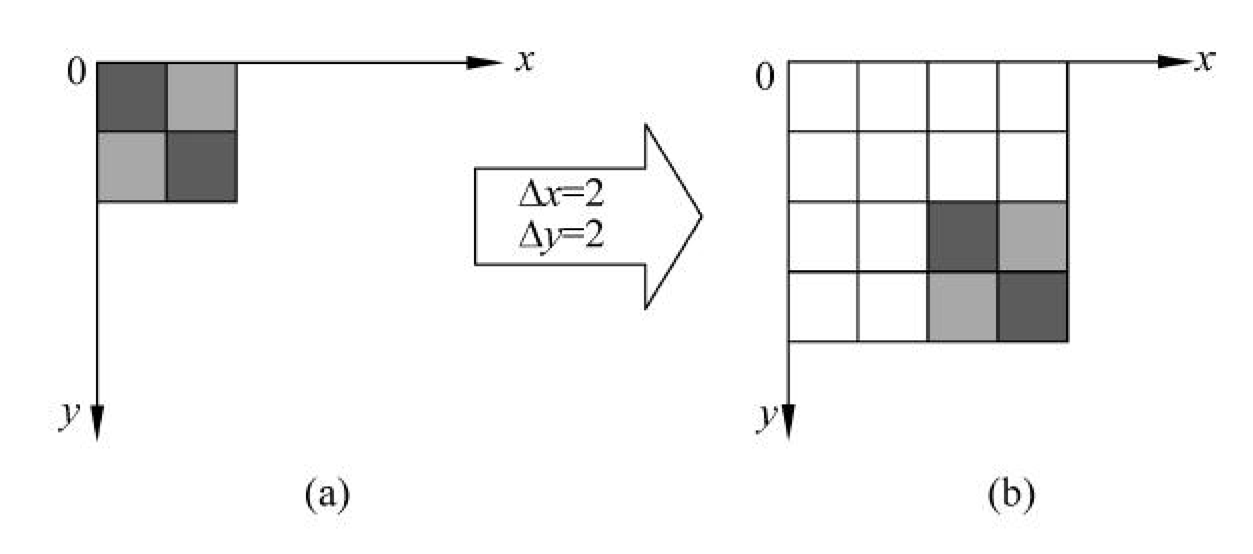


图 4-10 图像的平移

利用齐次坐标,点 $A(x,y)$的坐标可以表示为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{4-34}$$

相应地,也可以根据点 $A(x,y)$求解原始点 $A_0(x_0,y_0)$的坐标,即

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4-35}$$

显然,以上两个变换矩阵互为逆矩阵。

图像平移变换的特点是平移后的图像与原图像完全相同,平移后新图像上的每一点都可以在原图像中找到对应的点。以图 4-11 为例,对于新图中的像素点(0,0),代入式(4-35)可以求出对应原始图像中的像素(−2,−2),该点不在原始图像中。对于不在原图像中的点,可以直接将它的像素值统一设置为 0 或者 255,对于灰度图像则为黑色或白色。反之,若某像素点不在新图像中,同样说明原始图像中有某些像素点被移出了显示区域。如图 4-11 所示是图像平移变换的示例。图 4-11(a)是平移前的原始图像,图 4-11(b)是水平、垂直方向都平移 100 个像素后的图像,经过平移之后,原始图像的一些像素点被移出了显示区域,若想保留全部图像,应扩大新图像的显示区域。



(a) 平移前的图像



(b) 平移后的图像

图 4-11 图像的平移

4.4.3 比例缩放

通常情况下，数字图像的比例缩放是指将给定的图像在 x 方向和 y 方向按相同的比例缩放 a 倍，从而获得一幅新的图像，又称为全比例缩放。如果 x 方向和 y 方向缩放的比率不同，则图像的比例缩放会改变原始图像像素间的相对位置，产生几何畸变。设原始图像中的点 $A_0(x_0,y_0)$ 比例缩放后，在新图像中的对应点为 $A_1(x_1,y_1)$，则 $A_0(x_0,y_0)$ 和 $A_1(x_1,y_1)$ 之间坐标关系可表示为

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{4-36}$$

即

$$\begin{cases} x_1 = ax_0 \\ y_1 = ay_0 \end{cases}$$

若比例缩放所产生的图像中的像素在原图像中没有相对应的像素点，就需要进行灰度值的插值运算。一般有以下两种插值处理方法。

(1) 直接赋值为和它最相近的像素灰度值，这种方法称为最邻近插值法（nearest neighbor interpolation），该方法的主要特点是简单、计算量很小，但可能会产生马赛克现象。

(2) 通过其他数学插值算法来计算相应像素点的灰度值，这类方法处理效果好，但运算量会有所增加。

在式(4-36)所表示的比例缩放中，若 $a>1$，则图像被放大；若 $a<1$，则图像被缩小。以 $a=1/2$ 为例，即图像被缩小为原始图像的一半。图像被缩小一半以后根据目标图像和原始图像像素之间的关系，有如下两种缩小方法。

第一种方法如图 4-12 所示，此时缩放前后图像间像素点的对应关系如下：

$$\text{缩小图像}\begin{cases} (0,0) & \longleftrightarrow & (0,0) \\ (0,1) & \longleftrightarrow & (0,2) \\ (0,2) & \longleftrightarrow & (0,4) \\ (0,3) & \longleftrightarrow & (0,6) \\ (1,0) & \longleftrightarrow & (2,0) \\ (1,0) & \longleftrightarrow & (4,0) \\ \vdots & & \vdots \\ (3,0) & \longleftrightarrow & (6,0) \\ (3,1) & \longleftrightarrow & (6,2) \\ (3,2) & \longleftrightarrow & (6,4) \\ (3,3) & \longleftrightarrow & (6,6) \end{cases}\text{原始图像}$$

依此类推，可以逐点计算缩小图像各像素点的值，图像缩小之后所承载的信息量为原始图像的 50%，即在原始图像上，按行优先的原则，对于所处理的行，每隔一个像素取一点，每

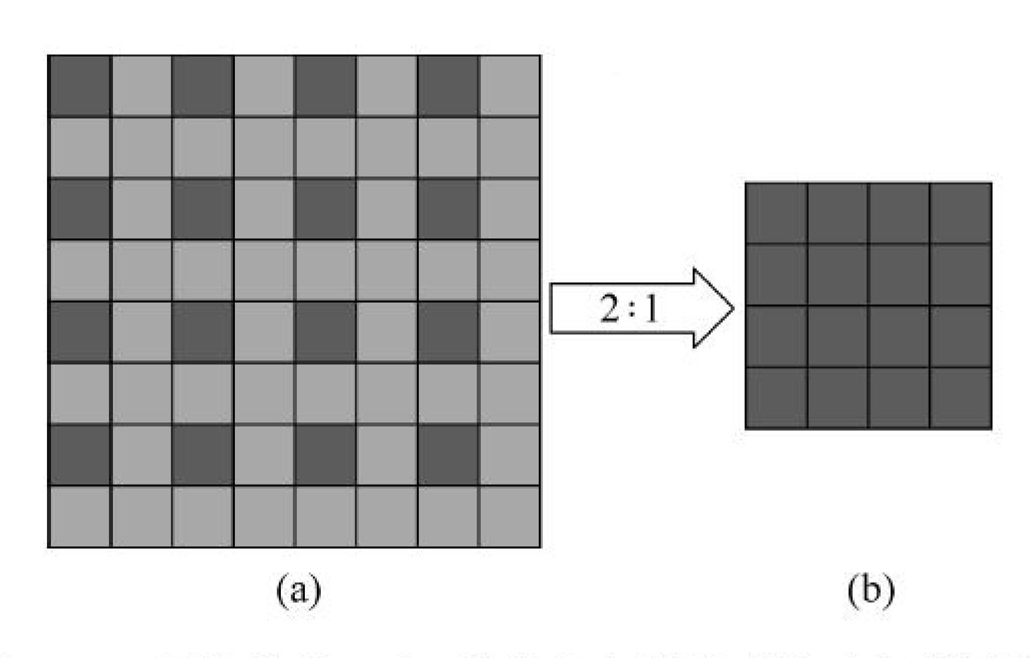


图 4-12　图像缩小一半(偶数行和偶数列构成新的图像)

隔一行进行操作。也就是说,取原始图像的偶数行和偶数列构成新的图像。

第二种方法是取原始图像奇数行和奇数列组成新图像,如图 4-13 所示。

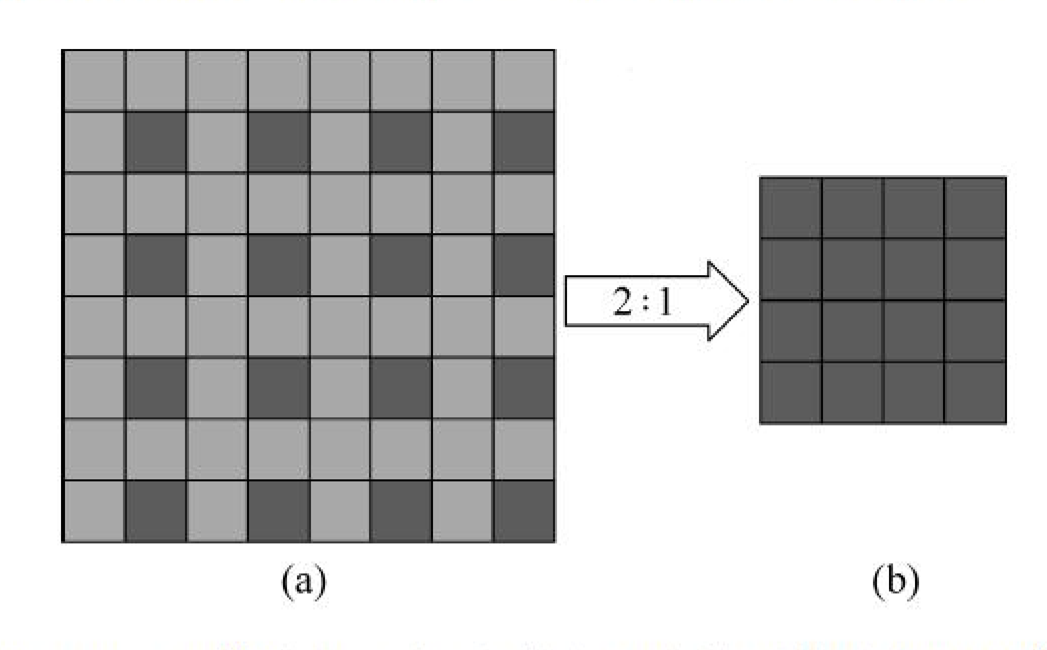


图 4-13　图像缩小一半(奇数行和奇数列构成新的图像)

如果图像按任意比例缩小,则以类似的方式按比例选择行和列上的像素点。若 x 方向与 y 方向缩放比例不同,这种变换将会使缩放以后的图像产生几何畸变。图像 x 方向和 y 方向的不同比例缩放的变换公式为

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}, \quad a \neq b \tag{4-37}$$

图像缩小变换是在已知的图像信息中以某种方式选择需要保留的信息。反之,图像的放大变换则需要对图像尺寸经放大后所多出来的像素点填入适当的像素值,这些像素点在原始图像中没有直接对应点,需要以某种方法进行估计。以 $a=b=2$ 为例,即原始图像按全比例放大 2 倍,实际上,这是将原始图像每行中各像素点重复取值一遍,然后每行重复一次。根据理论计算,放大以后图像中的像素点(0,0)对应于原始图中的像素点(0,0),(0,2)对应于原始图像中的(0,1);但放大后图像的像素点(0,1)对应于原始图中的像素点(0,0.5),(1,0)对应于原始图中的(0.5,0),原始图像中不存在这些像素点,那么放大图像如何处理这些问题呢?以像素点(0,0.5)为例,这时可以采取以下两种方法和原始图像对应,其余点依此类推。

(1) 将原始图中的像素点(0,0.5)近似为原始图像的像素点(0,0)。

(2) 将原始图中的像素点(0,0.5)近似为原始图像的像素点(0,1)。

以图 4-14 所示的一段直线为例来说明这两种放大的细微差别(为讨论问题方便,图中对原始图像和放大图像均已标注像素坐标值)。

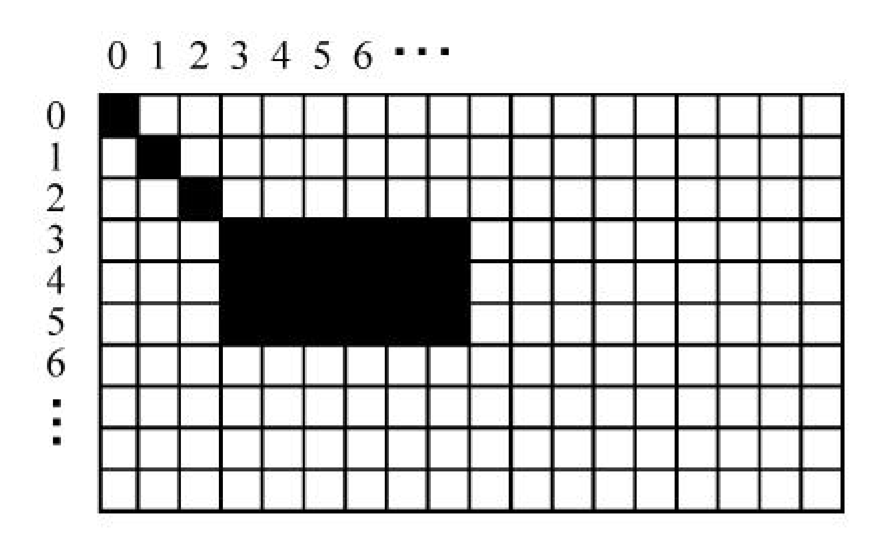
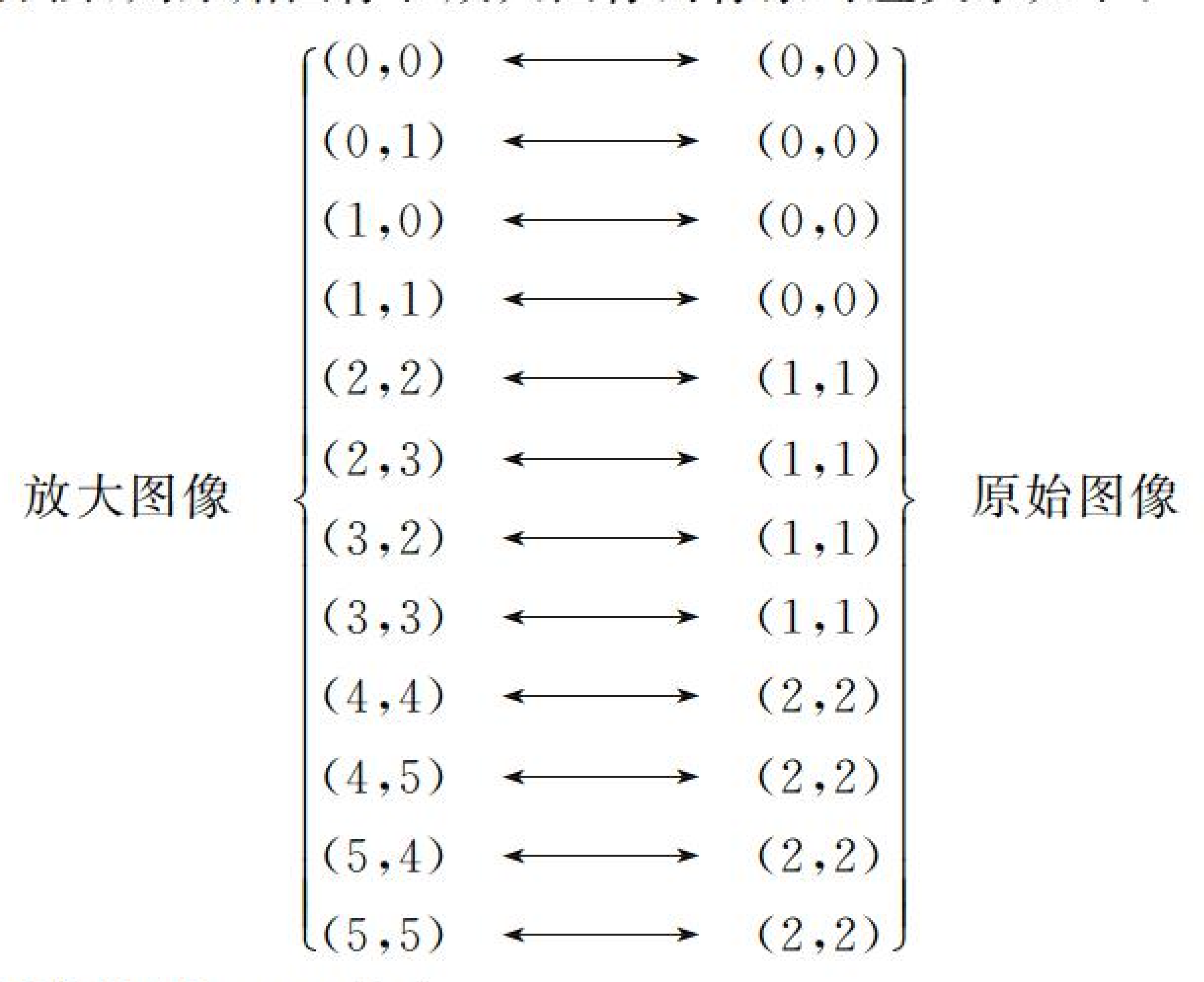


图 4-14 放大前的原始图像

若采取第一种方法,则原始图像和放大图像的像素对应关系如下:



其对应的放大图像如图 4-15 所示。

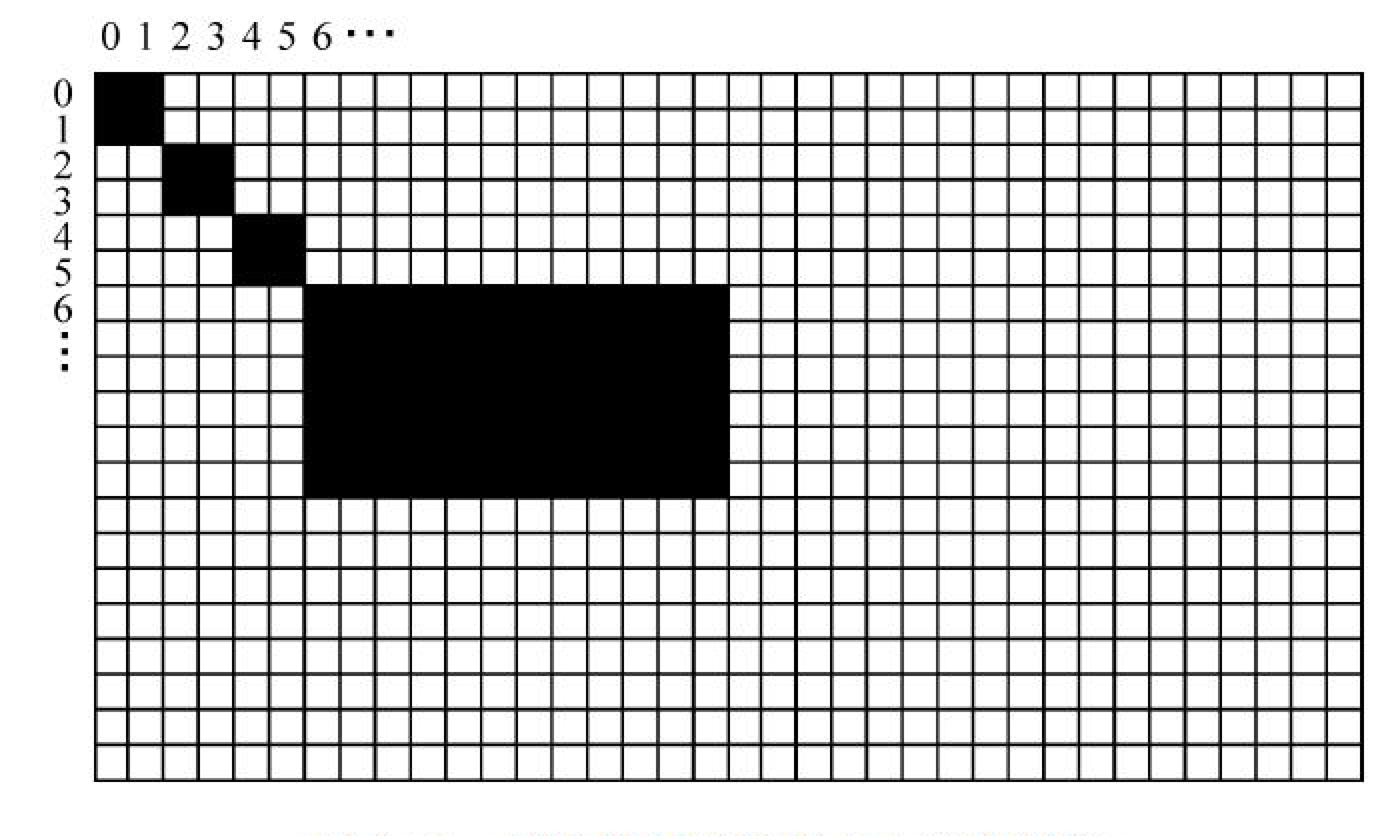
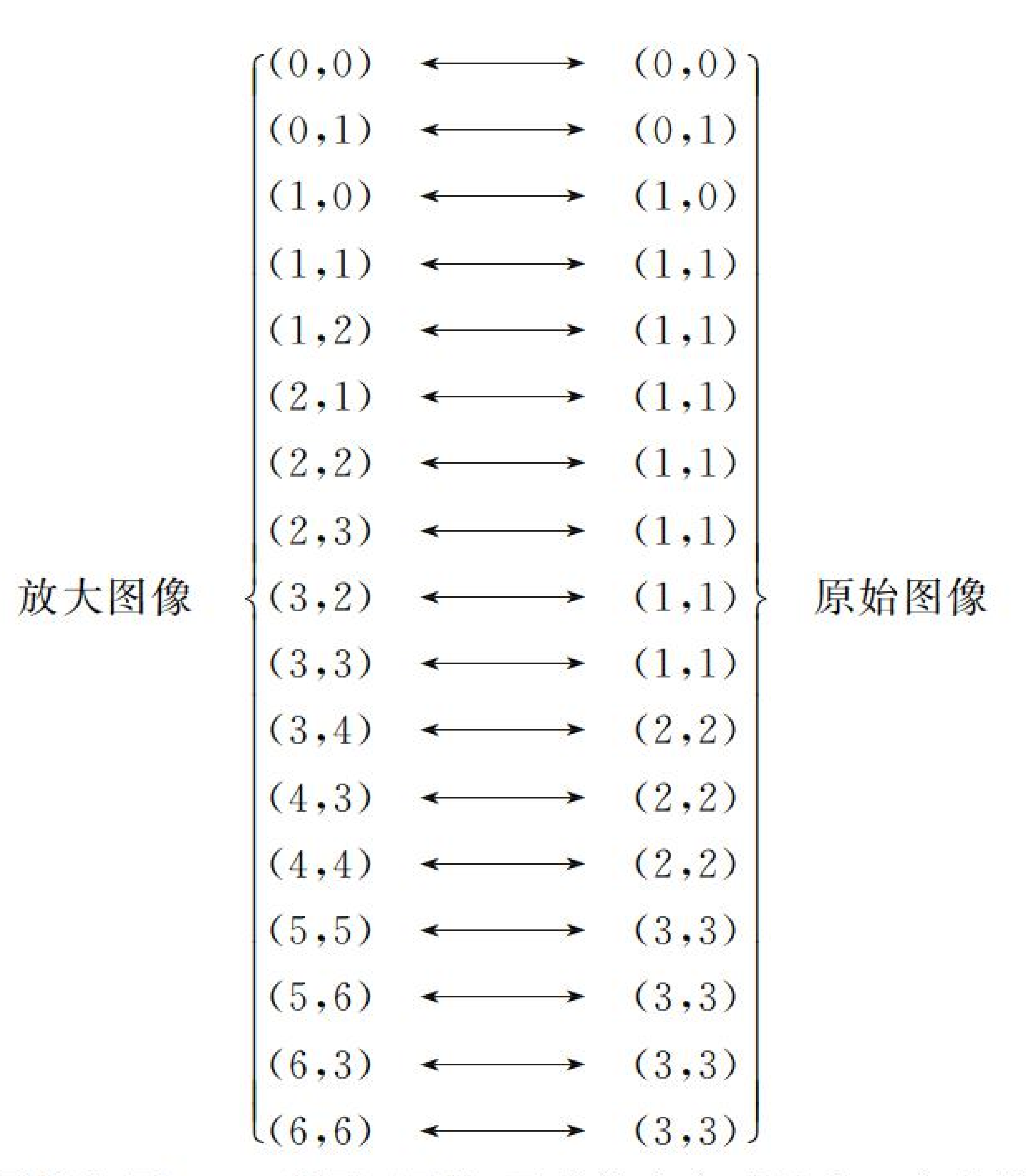


图 4-15 按最近邻域法放大 2 倍的图像

若采取第二种方法,则原始图像和放大图像的像素对应关系如下:



其对应的放大图像如图 4-16 所示，因此，两种放大方式具有一定差别。

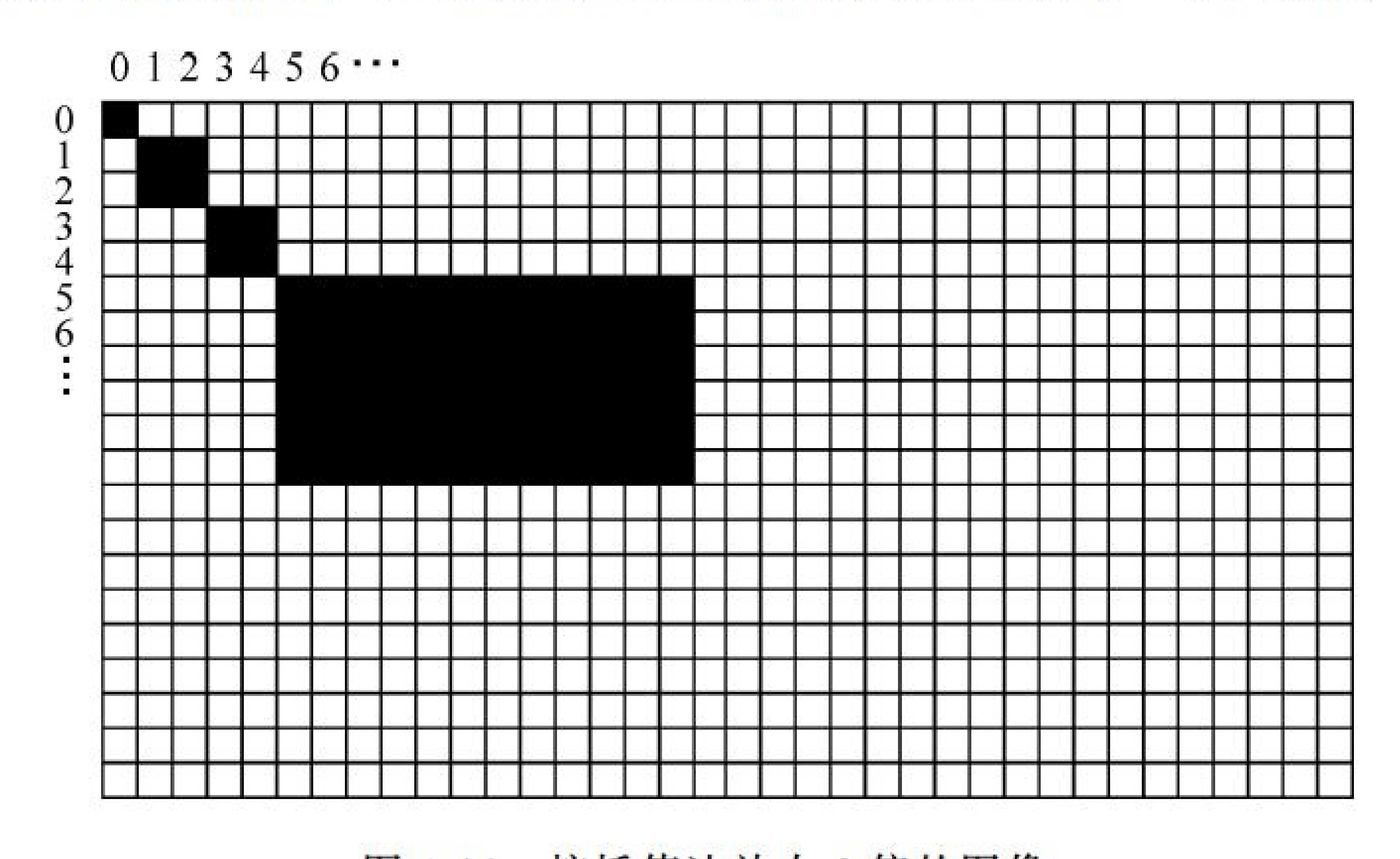


图 4-16 按插值法放大 2 倍的图像

一般地，按比例将原图像放大 a 倍时，如果按照最近邻域法，则需要将一个像素值添在新图像的 $a \times a$ 的方块中，如图 4-17 所示。因此，如果放大倍数过大，按照这种方法填充灰度值会出现马赛克效应。为了避免马赛克效应，提高几何变换后的图像质量，可以采用不同复杂程度的线性插值法填充放大后所多出来的相关像素点的灰度值。

4.4.4 镜像变换

镜像变换也是与人们日常生活密切相关的一种变换，图像的镜像(mirror)变换不改变图像的形状。图像的镜像变换包括水平镜像和垂直镜像两种。图像的水平镜像变换是将图像左半部分和右半部分以图像垂直中轴线为中心进行镜像对换，如图 4-18(b)所示；图像的

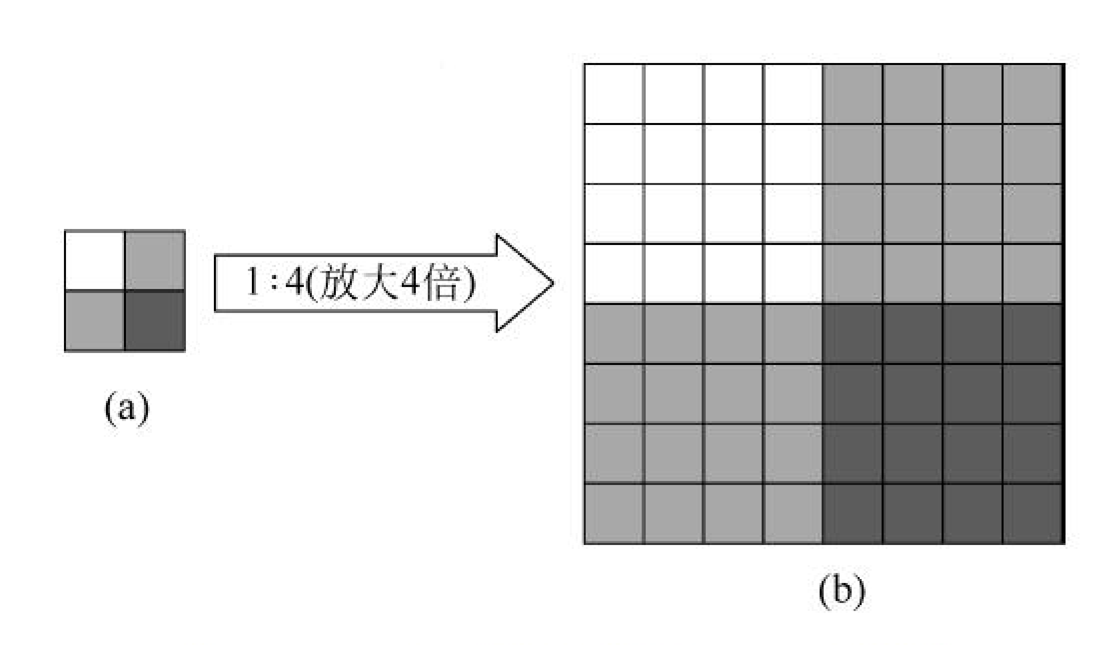


图 4-17　按最近邻域法放大 5 倍的图像

垂直镜像变换是将图像上半部分和下半部分以图像水平中轴线为中心进行镜像对换，如图 4-18(c)所示。

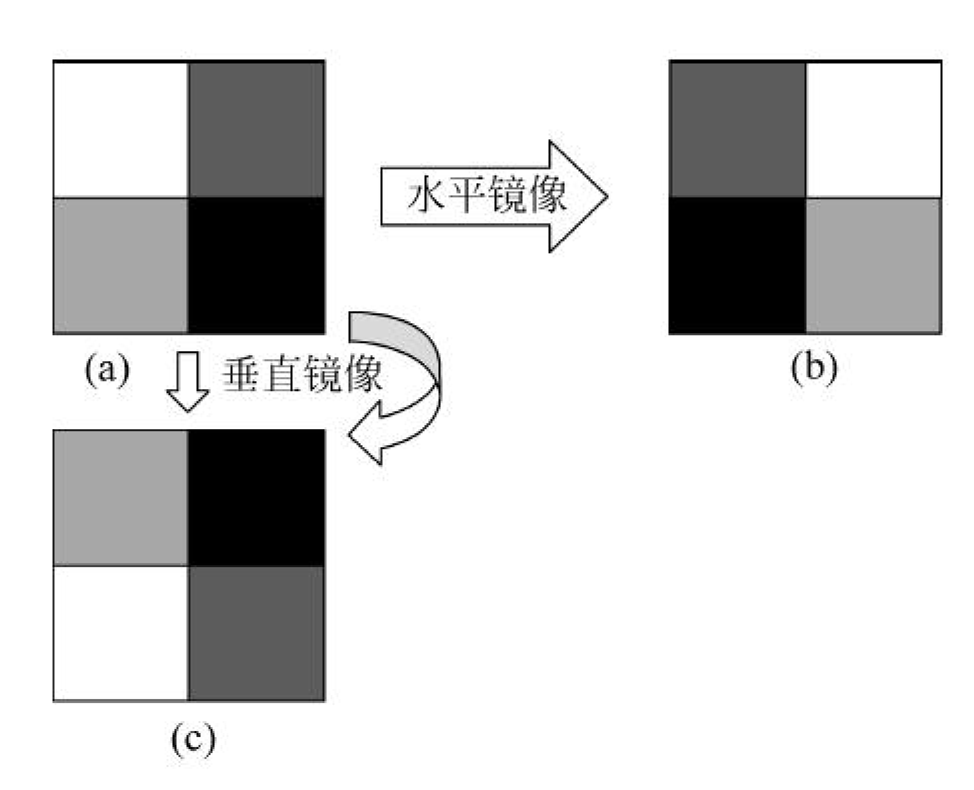


图 4-18　图像的镜像位置关系

1. 水平镜像

设点 $A_0(x_0, y_0)$进行镜像后的对应点为 $A(x,y)$，图像高度为 h，宽度为 w，原始图像中的点 $A_0(x_0, y_0)$经过水平镜像后坐标将变为

$$\begin{cases} x = w - x_0 \\ y = y_0 \end{cases} \tag{4-38}$$

图像的水平镜像变换用矩阵形式表示为

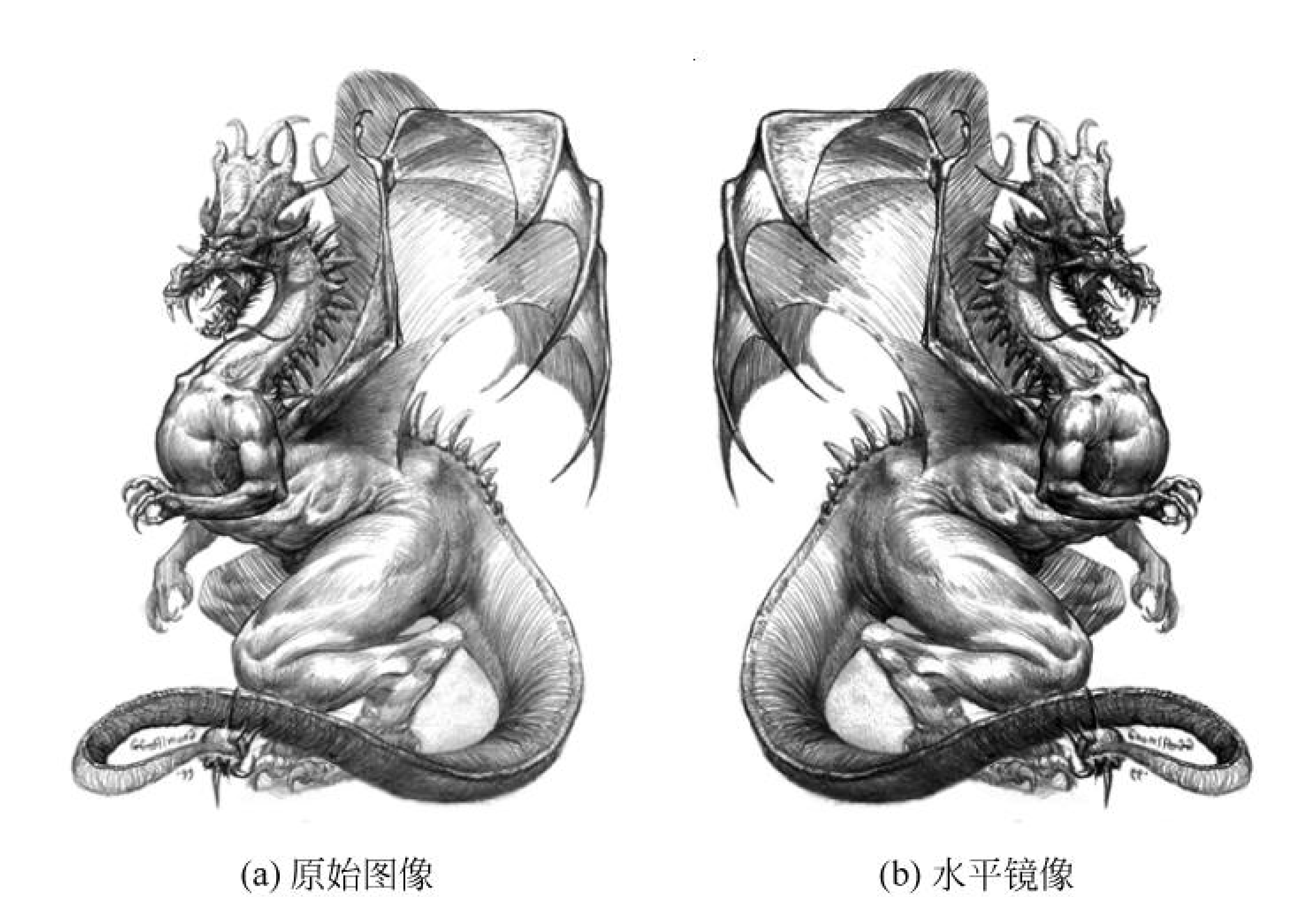
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{4-39}$$

同样也可以根据点 $A(x,y)$求解原始点 $A_0(x_0, y_0)$的坐标，矩阵表示形式为

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4-40}$$

例 4-5：图像水平镜像

如图 4-19 所示为图像水平镜像的变换示例。图 4-19(a)为原始图像，图 4-19(b)为原始图像经水平镜像变换以后的图像。



(a) 原始图像　　　　(b) 水平镜像

图 4-19　图像的水平镜像

2. 垂直镜像

对于垂直镜像变换，设点 $A_0(x_0, y_0)$经过垂直镜像后将变为点 $A(x, y)$，原始图像中的点 $A_0(x_0, y_0)$经过垂直镜像后坐标将变为

$$\begin{cases} x = x_0 \\ y = h - y_0 \end{cases} \tag{4-41}$$

图像的垂直镜像变换也可以用矩阵变换表示，其矩阵表示形式为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{4-42}$$

垂直镜像也可以根据点 $A(x, y)$求解原始点 $A_0(x_0, y_0)$的坐标，矩阵表示形式为

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4-43}$$

例 4-6：图像垂直镜像

如图 4-20 所示为图像垂直镜像变换示例。图 4-20(a)为原始图像，图 4-20(b)为原始图像经垂直镜像变换以后的图像。

4.4.5　旋转变换

图像的旋转变换是几何学中研究的重要内容之一。一般情况下，图像的旋转变换是指以图像的中心为原点，将图像上的所有像素都旋转同一个角度的变换。图像经过旋转变换之后，图像的位置发生了改变，但旋转后，图像的大小一般会改变。和平移变换一样，在图像旋转变换中既可以把转出显示区域的图像截去，也可以扩大显示区域的图像范围以显示图像的全部。


(a) 原始图像


(b) 垂直镜像

图 4-20　图像的垂直镜像

设原始图像的任意点 $A_0(x_0,y_0)$ 经旋转 β 角度以后到新的位置 $A(x,y)$，为表示方便，采用极坐标形式表示，原始点的角度为 α，如图 4-21 所示。

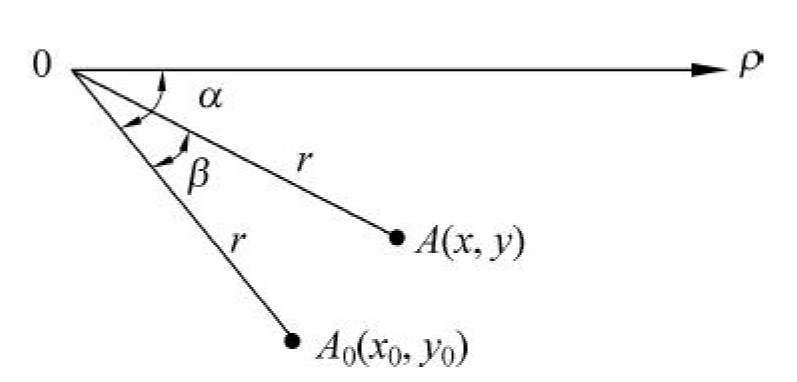


图 4-21　图像旋转

根据极坐标与二维垂直坐标的关系，原始图像的点 $A_0(x_0,y_0)$ 的 x 和 y 坐标为

$$\begin{cases} x_0 = r\cos\alpha \\ y_0 = r\sin\alpha \end{cases} \tag{4-44}$$

旋转到新位置以后点 $A(x,y)$ 的坐标为

$$\begin{cases} x = r\cos(\alpha-\beta) = r\cos\alpha\cos\beta + r\sin\alpha\sin\beta \\ y = r\sin(\alpha-\beta) = r\sin\alpha\cos\beta - r\cos\alpha\sin\beta \end{cases} \tag{4-45}$$

由于旋转变换需要以点 $A_0(x_0,y_0)$ 表示点 $A(x,y)$，因此对上式进行简化，可得出

$$\begin{cases} x = x_0\cos\beta + y_0\sin\beta \\ y = -x_0\sin\beta + y_0\cos\beta \end{cases} \tag{4-46}$$

同样，图像的旋转变换也可以用矩阵形式表示为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{4-47}$$

图像旋转之后也可以根据点 $A(x,y)$ 求解原始点 $A_0(x_0,y_0)$ 的坐标，其矩阵表示形式为

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4-48}$$

图像旋转之后，由于数字图像的坐标值必须是整数，因此，可能引起图像部分像素点的位置的小变化，因此，这时图像的边缘、图像内部会发生一定的改变，从严格的角度分析，由于扩充了新的空白像素点，图像的大小也会发生一定的改变。但一般情况下，这种变化可能

感觉不明显。

例 4-7：图像旋转产生的变化

如图 4-22 示例了图像旋转所产生的影响。原图像经过了两次 45°和 135°旋转变换，旋转 360°之后，图 4-22(b)的字迹发生了较明显的变化，特别是字体的边缘产生的变化更为明显。

旋转
变换

(a) 原始图像



(b) 旋转以后的图像

图 4-22　旋转变换产生的影响

若图像旋转角 $\beta=45°$时，则变换关系为

$$\begin{cases} x = 0.707x_0 + 0.707y_0 \\ y = -0.707x_0 + 0.707y_0 \end{cases} \tag{4-49}$$

以原始图像的点(1,1)为例，旋转以后，均为小数，经舍入以后为(1,0)，产生了位置误差。因此，图像旋转以后可能会发生一些细微变化。

对图像进行旋转变换时应注意以下几点：

(1) 为了避免图像旋转之后可能产生的信息丢失，可以先进行平移，然后进行图像旋转。

(2) 图像旋转之后，可能会出现一些空白点，需对这些空白点进行灰度级的插值处理，否则影响旋转之后的图像质量。

在某些情况下，一个几何变换需要两个独立算法支持。其中一个算法来实现空间变换本身，用它描述每个像素点如何从其初始位置移动到目标位置，即每个像素点的几何变换；而另一个算法用于灰度级的插值，这是因为经过变换之后，图像的大小以及像素的坐标可能发生变化，因而需要进行灰度级插值来提高图像质量，反之亦然。

需要指出的是，上述所讨论的旋转是绕坐标轴原点(0,0)进行的，如果图像旋转是绕一个其他指定点(a,b)旋转，则先要将坐标系平移到该点，再进行旋转，然后将旋转后的图像平移回原来的坐标原点。

4.4.6　复合变换

图像的复合变换是指对给定图像进行两次或两次以上的平移、镜像、比例、旋转等基本变换的多次变换，又称为级联变换。由于引入齐次坐标后图像的基本变换采用了统一的矩阵表示形式，根据矩阵理论可知，对给定图像按顺序连续进行多次基本图像变换，其变换的矩阵仍然可以用 3×3 矩阵表示。复合变换的矩阵等于基本变换的矩阵按变换顺序依次相乘。

若对图像依次进行了 n 次平移、镜像、比例、旋转等基本变换，其变换矩阵分别为 $\boldsymbol{T}_1$，

$\boldsymbol{T}_2,\cdots,\boldsymbol{T}_n$,则 n 次变换之后的复合变换矩阵 $\boldsymbol{T}$ 可以表示为

$$\boldsymbol{T}=\boldsymbol{T}_1\boldsymbol{T}_2\cdots\boldsymbol{T}_{n-1}\boldsymbol{T}_n \tag{4-50}$$

根据复合变换的组合类型,复合变换可以分为如下两类。

(1) 同类型复合变换。复合变换由同一种基本变换组成,即相同的基本变换连续进行多次,例如,复合平移、复合比例缩放、复合旋转等。

(2) 不同类型复合变换。复合变换由不同类型的基本变换组成,即不同的基本变换连续进行多次,例如,图像的转置、绕任意点的比例缩放、绕任意点的旋转等。

现对同类型的复合变换讨论如下。

1. 复合平移

以包含两次基本平移的复合平移为例,指将图像先平移到新的位置后 $A_1(x_1,y_1)$,再将图像平移到的位置 $A_2(x_2,y_2)$,复合平移矩阵为

$$\boldsymbol{T}=\boldsymbol{T}_1\boldsymbol{T}_2=\begin{bmatrix}1&0&\Delta x_1\\0&1&\Delta y_1\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&\Delta x_2\\0&1&\Delta y_2\\0&0&1\end{bmatrix}=\begin{bmatrix}1&0&\Delta x_1+\Delta x_2\\0&1&\Delta y_1+\Delta y_2\\0&0&1\end{bmatrix} \tag{4-51}$$

观察复合变换矩阵 $\boldsymbol{T}$ 可以发现,两次平移之后,平移的距离等于两次平移距离之和。因此,采用齐次坐标表示几何变换,不仅形式简洁,而且具有明显的几何意义。两次以上的复合平移变换可由此类推。

2. 复合比例缩放

同样,以包含两次比例缩放变换的复合缩放变换为例,复合比例矩阵为

$$\boldsymbol{T}=\boldsymbol{T}_1\boldsymbol{T}_2=\begin{bmatrix}a_1&0&0\\0&b_1&0\\0&0&1\end{bmatrix}\begin{bmatrix}a_2&0&0\\0&b_2&0\\0&0&1\end{bmatrix}=\begin{bmatrix}a_1a_2&0&0\\0&b_1b_2&0\\0&0&1\end{bmatrix} \tag{4-52}$$

对图像连续进行多次比例缩放变换,最后的复合变换矩阵只需要对两次变换的比例常数进行乘积运算即可。

3. 复合旋转

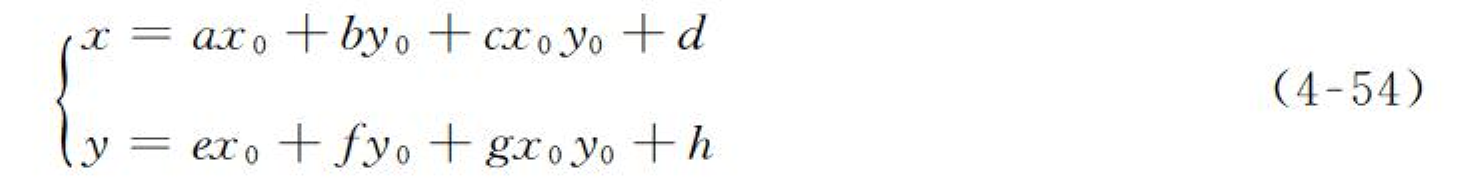
类似地,对图像连续进行多次旋转变换,最后合成的旋转变换矩阵等于各次旋转角度之和。以包含两次旋转变换的复合旋转变换为例,其最后的变换矩阵为

$$\boldsymbol{T}=\boldsymbol{T}_1\boldsymbol{T}_2=\begin{bmatrix}\cos\beta_1&\sin\beta_1&0\\-\sin\beta_1&\cos\beta_1&0\\0&0&1\end{bmatrix}\begin{bmatrix}\cos\beta_2&\sin\beta_2&0\\-\sin\beta_2&\cos\beta_2&0\\0&0&1\end{bmatrix}=\begin{bmatrix}\cos(\beta_1+\beta_2)&\sin(\beta_1+\beta_2)&0\\-\sin(\beta_1+\beta_2)&\cos(\beta_1+\beta_2)&0\\0&0&1\end{bmatrix} \tag{4-53}$$

以上均为相对于原点(图像中心)进行比例、旋转等复合变换,如果要相对其他参考点进行以上变换,则要先进行平移,然后再进行其他基本变换,最后形成图像的复合变换。不同的复合变换,所包含的基本变换的数量和次序各不相同,但是无论其变换过程多么复杂,都可以分解成若干基本变换,都可以采用齐次坐标表示,且图像复合变换矩阵由一系列基本变换矩阵依次相乘而得到。

4.4.7 控制点变换

控制点变换是通过测定若干特定坐标点的位移量来确定坐标变换方程的系数的方法。如图 4-23 所示,若已知输入图像的 1、2、3、4 四个点与输出图像上 A、B、C、D 四个点对应(即具有四个控制点对的位置对应关系),则可以根据这四对已知控制点对,通过如下代数变换方程确定几何变换关系。



$$\begin{cases} x = ax_0 + by_0 + cx_0y_0 + d \\ y = ex_0 + fy_0 + gx_0y_0 + h \end{cases} \tag{4-54}$$

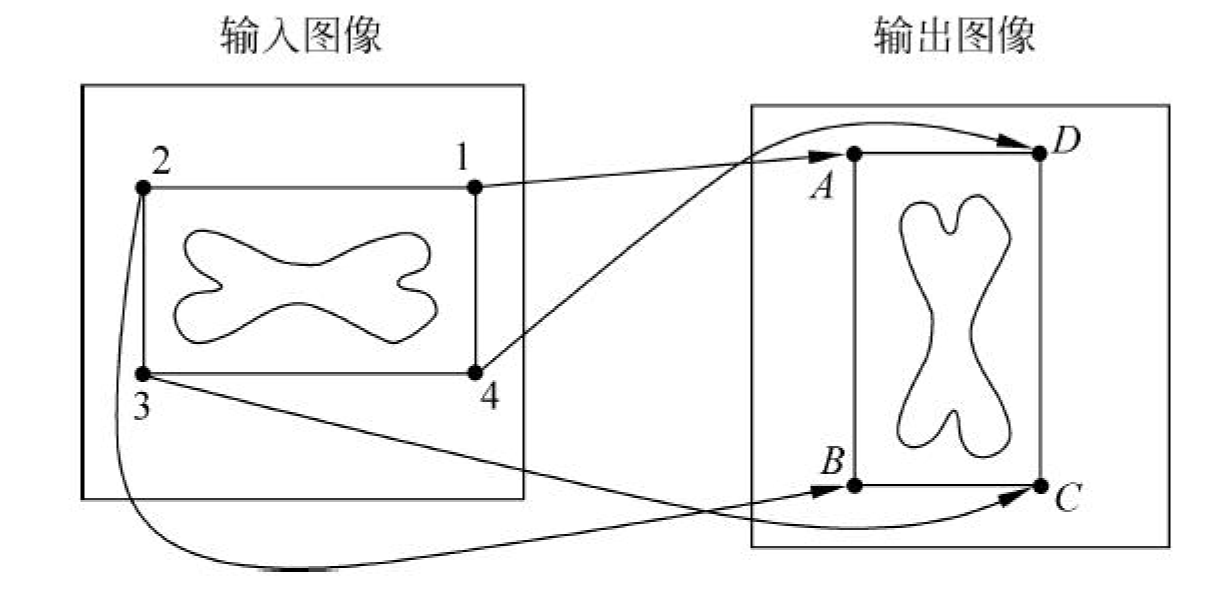


图 4-23 控制点位置映射关系

求解上述变换方程中的系数,就可以得到满足要求的近似变换关系,即可确定所有落入矩形框内的输出点。实际上,上述变换可以表示为更一般的代数幂函数方程变换形式:

$$\begin{cases} x = \sum_{i=0}^{N} \sum_{j=0}^{N-1} a_{ij} x_0^i y_0^j \\ y = \sum_{i=0}^{N} \sum_{j=0}^{N-1} b_{ij} x_0^i y_0^j \end{cases} \tag{4-55}$$

若 N 取值增大,为确定变换系数所需的对应控制点对数量就增多,变换精度就越高。多数情况下,N 取值在 2～3 之间即可满足精度要求。控制点变换关系常用于几何畸变的校正。

4.4.8 透视变换

在光线的照射下,三维物体可以在二维平面上形成投影,这种将三维物体或对象转变为二维图形表示的过程称为投影变换,投影中心称为视点。根据视点与投影平面之间距离的不同,投影可分为平行投影和透视投影。平行投影的视点与投影平面之间的距离为无穷大,若该距离是有限的则称为透视投影(变换)。

透视投影即透视变换,透视投影的距离决定着透视投影的透视缩小效应,三维物体

或对象透视投影的大小与物体到视点的距离成反比。例如,平行于投影面且长度相等的两段直线,离投影中心距离较近的线段,其透视投影长,离投影中心远的线段,其透视投影短。这种效应所产生的视觉效果与照相机系统和人的视觉系统十分类似。与平行投影相比,透视投影的深度感更强,看上去更真实,但透视投影不能真实地反映物体的精确尺寸和形状。

透视投影时,平行于投影面的平行线的投影依然平行,而不平行于投影面的平行线的投影则会聚集到一个点,这个点称为灭点(vanishing point)。灭点可以看作是无限远处的一点在投影面上的投影。透视投影的灭点可以有无限多个,不同方向的平行线在投影面上就可以形成不同的灭点,其中坐标轴方向的平行线在投影面上所形成的灭点称为主灭点。对于三维空间(xyz 三个坐标轴)主灭点最多有 3 个。透视投影一般根据主灭点的数量(即按投影面与坐标轴的夹角)进行分类,因此可分为一点透视、二点透视和三点透视,如图 4-24 所示。

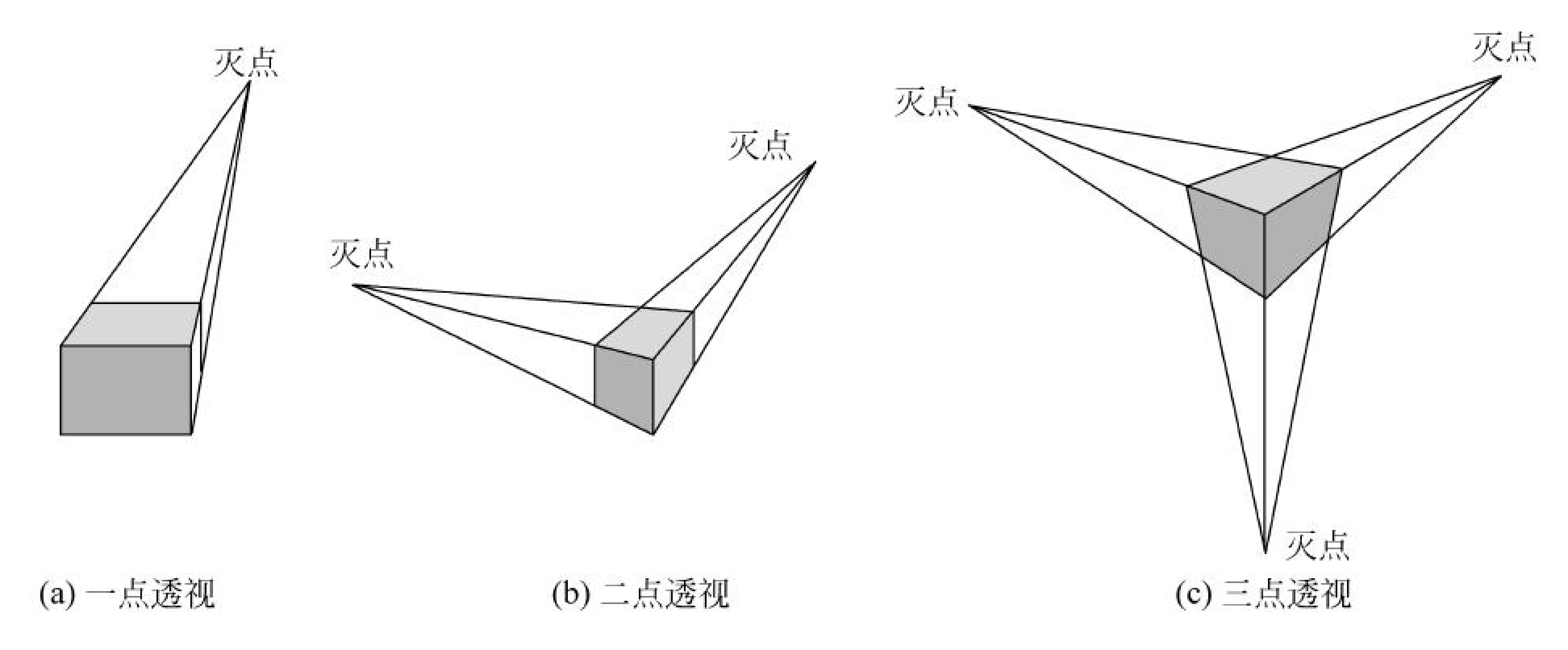


图 4-24 透视变换的类型

以一点透视为例,如图 4-24(a)所示,它仅有一个主灭点,即投影面与一个坐标轴正交,与另外两个坐标轴平行。进行一点透视投影变换时,应很好地考虑投影的布局,以避免三维形体或对象的平面域积聚成直线或直线积聚成点而影响直观性。主要考虑如下几点:

(1) 三维物体或对象与画面(投影面)的相对位置。

(2) 视距,即视点与画面的距离。

(3) 视点的高度。

如图 4-25 所示,设视点在坐标原点,z 坐标轴方向与观察方向重合一致,点 $P(x,y,z)$ 为三维形体或对象上的一点,经一点透视变换后在投影面(观察平面)UOV 上的对应点为 $P(x',y',z')$,投影面与视点的距离为 d,并与 z 轴垂直,z 轴过投影面窗口的中心,窗口是边长为 $2a$ 的正方形,根据几何学知识可得

$$\begin{cases}\dfrac{x'}{x}=\dfrac{y'}{y}=\dfrac{z'}{z}=k \\ z'=d\end{cases} \tag{4-56}$$

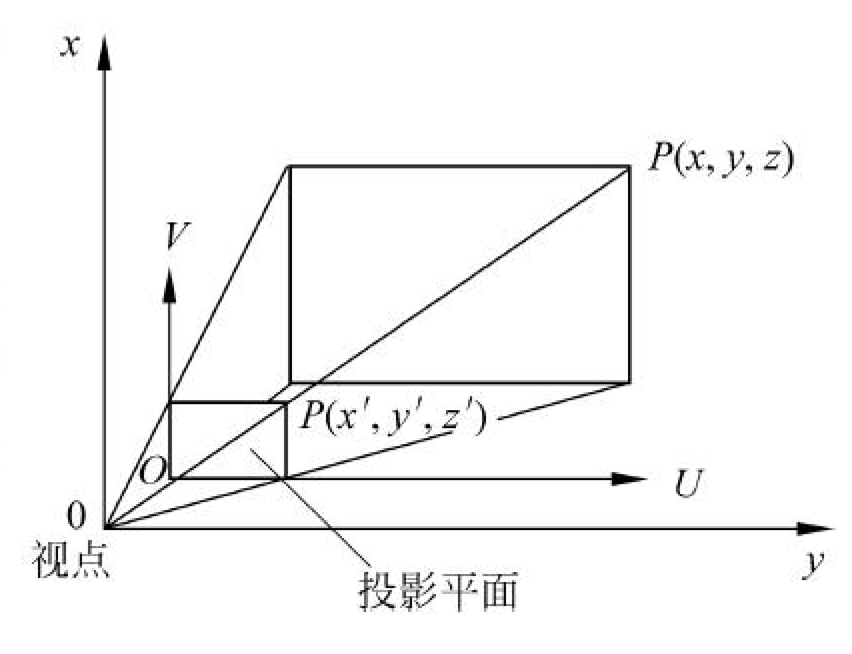


图 4-25 一点透视变换原理

利用齐次坐标，与二维几何变换类似，将该过程写成变换矩阵形式为

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4-57}$$

这就是图像的透视变换——一点透视变换，即

$$[x' \quad y' \quad z' \quad 1] = \left[\frac{x}{z}d \quad \frac{y}{z}d \quad d \quad 1\right]$$

实际上，更一般情况的一点透视变换矩阵也可以用一个 4×4 的矩阵表示。

4.4.9 其他变换

如前所述，齐次坐标为表示各种基本变换及其复合变换提供了一个简单有效的方法。然而，在许多图像处理与分析的具体应用中，所需的几何变换都相当复杂，甚至有些无法用简便的数学式来表达。这时几何变换常常要通过对实际图像进行测量才可获得，因此，也可以采用测量结果而不是函数形式描述几何变换。例如，在对由摄像机拍摄的具有几何畸变的图像进行几何校正时，首先应将一个矩形栅格目标数字化并显示出来。因为摄像机有几何变形，所显示的图案不一定是准确的矩形，因此所求几何变换应能使其栅格图案再次被复原为准确的矩形，从而修正摄像机所产生的畸变。一般而言，采用同样的几何变换可用于校正同一摄像机产生的几何畸变。

图像几何变换的一个重要应用是消除由于摄像机导致的数字图像的几何畸变。当需要从数字图像中得到定量的空间测量数据时，几何校正被证明是十分重要和有效的，特别是对于从卫星遥感图像或飞机上通过侧视雷达所得到的图像，一般均产生了比较严重的几何畸变。因此，这些图像都需要先经过几何校正，然后才能对其内容做出解释。

几何校正是遥感图像预处理中必须解决的问题，遥感影像的几何畸变分为内部畸变和外部畸变。内部畸变是由于传感器内在的因素所引起的畸变，如镜头的像差、比例尺变化、歪斜、中心移动、扫描非线性、辐射状畸变和扭曲等。外部畸变是由于运载工具的姿态和目标引起的畸变，包括飞行器飞行高度变化引起的比例尺误差，由于飞行器的俯仰、滚摆及偏航所引起的畸变，地形起伏的畸变，地球曲率引起的畸变等。

总之，遥感仪器在获取地球表面信息的过程中，受到各种内部的和外部的干扰，使所获得的图像包含着各种辐射畸变和几何畸变。无论对于航空遥感还是卫星遥感，这些畸变和误差都必须予以纠正。对于仪器内在因素而导致的误差，可根据预先测试的标定数据进行校正。有时这种标定数据还需要进行修正，以适应仪器在长期运行中可能产生的漂移和变化。外部畸变的校正，则应根据不同的情况进行处理。

对于遥感图像的几何畸变，可以利用地面控制点方法进行几何校正，即利用地面上的水陆交叉点、公路交叉点、河流水域分叉点等具有明显特征的地点，以及这些特征点的已知位置或地面高度等参数，并通过这些控制点的对应关系进行几何校正。例如，对于陆地卫星图像的几何校正，可以通过适当数量（N 点）的地面控制点，按如下关系进行几何畸变校正。

$$\begin{cases} u_i = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \\ v_i = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 \end{cases} \quad i = 1,2,3,\cdots,N-1 \tag{4-58}$$

式中，(u,v)是校正前原始图像控制点的坐标，(x,y)是校正后目标图像对应点的坐标，令

$$\boldsymbol{u}=\begin{bmatrix}u_1\\u_2\\\vdots\\u_N\end{bmatrix}\quad \boldsymbol{v}=\begin{bmatrix}v_1\\v_2\\\vdots\\v_N\end{bmatrix}$$

$$\boldsymbol{a}=\begin{bmatrix}a_1\\a_2\\\vdots\\a_5\end{bmatrix}\quad \boldsymbol{b}=\begin{bmatrix}b_1\\b_2\\\vdots\\b_5\end{bmatrix}$$

$$\boldsymbol{A}=\begin{bmatrix}1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2\\1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2\\\cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\1 & x_N & y_N & x_N^2 & x_Ny_N & y_N^2\end{bmatrix}$$

当对应的控制点数多于 6 个时，将产生过约束，以均方误差最小求解 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，令

$$\boldsymbol{E}=(\boldsymbol{u}-\boldsymbol{Aa})^{\mathrm{T}}(\boldsymbol{u}-\boldsymbol{Aa})+(\boldsymbol{v}-\boldsymbol{Ab})^{\mathrm{T}}(\boldsymbol{v}-\boldsymbol{Ab})$$

根据最小二乘方法，可得

$$\begin{cases}\boldsymbol{a}=(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\boldsymbol{u}\\\boldsymbol{b}=(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\boldsymbol{v}\end{cases}$$

可以根据校正精度的要求选择变换多项式。一般情况下，三次以下多项式可以满足陆地卫星图像的几何畸变校正要求。此外，还有一些图像系统采用诸如极坐标、柱坐标、球面坐标等非矩形的像素坐标，用普通图像显示设备观察这些图像时，也需先对图像进行校正，即应将图像转换为矩形像素坐标。例如，在油井套管缺陷识别中，有时需要将极坐标系中的内窥镜图像转换为直角坐标系中的图像，然后进行分析与处理。

几何变换的另一个应用是对相似的图像进行配准，以便进行图像比较，典型的应用是利用图像相减来检测运动或变化。有时，为便于解释需将图像以另一种样式表示，这时也会用到几何变换，地图绘制中的图像投影也会用到几何变换。例如，在利用从宇宙飞船传回来的图像拼成地球、月球及行星的航拍镶嵌地图时就需应用几何变换。

4.5 灰度级插值

在进行图像的比例缩放、图像的旋转及其复合变换时，原始图像的像素坐标(x,y)为整数，而变换后目标图像的位置坐标并非整数，反过来也是如此。因此，在进行图像的几何变换时，除了要进行几何变换运算之外，还需要进行灰度级插值处理。常用的灰度插值方法有三种——最近邻法、双线性插值法和三次内插法。

4.5.1 最近邻法

最近邻法是一种简单的插值方法，如图 4-26 所示，它是通过计算与点 $P(x_0,y_0)$临近的四个点，并将与点 $P(x_0,y_0)$最近的整数坐标点(x,y)的灰度值取为 $P(x_0,y_0)$点灰度近似

值。在 $P(x_0,y_0)$ 点各相邻像素间灰度变化较小时，这种方法是一种简单快速的方法，但当 $P(x_0,y_0)$ 点相邻像素间灰度值差异很大时，这种灰度估值方法会产生较大的误差，甚至可能影响图像质量。

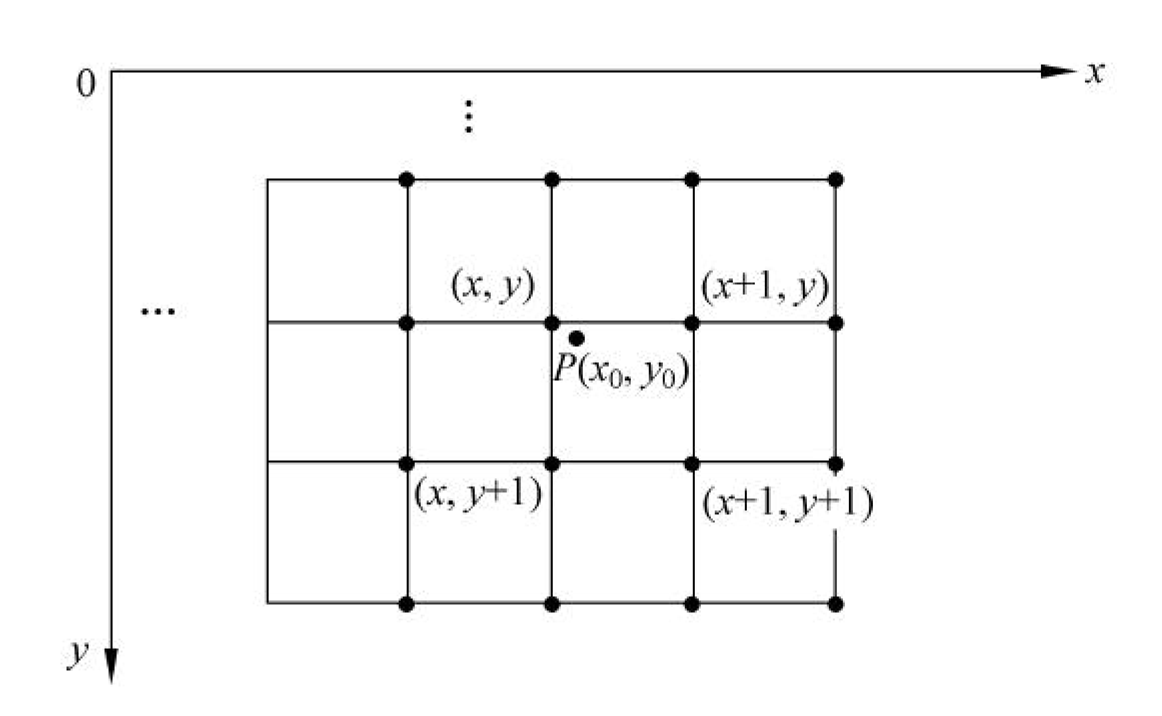


图 4-26　最近邻法插值示意图

4.5.2　双线性插值法

双线性插值法是对最近邻法的一种改进，即用线性内插方法，根据点 $P(x_0,y_0)$ 的四个相邻点的灰度值，通过两次插值计算出灰度值 $f(x_0,y_0)$，如图 4-27 所示。

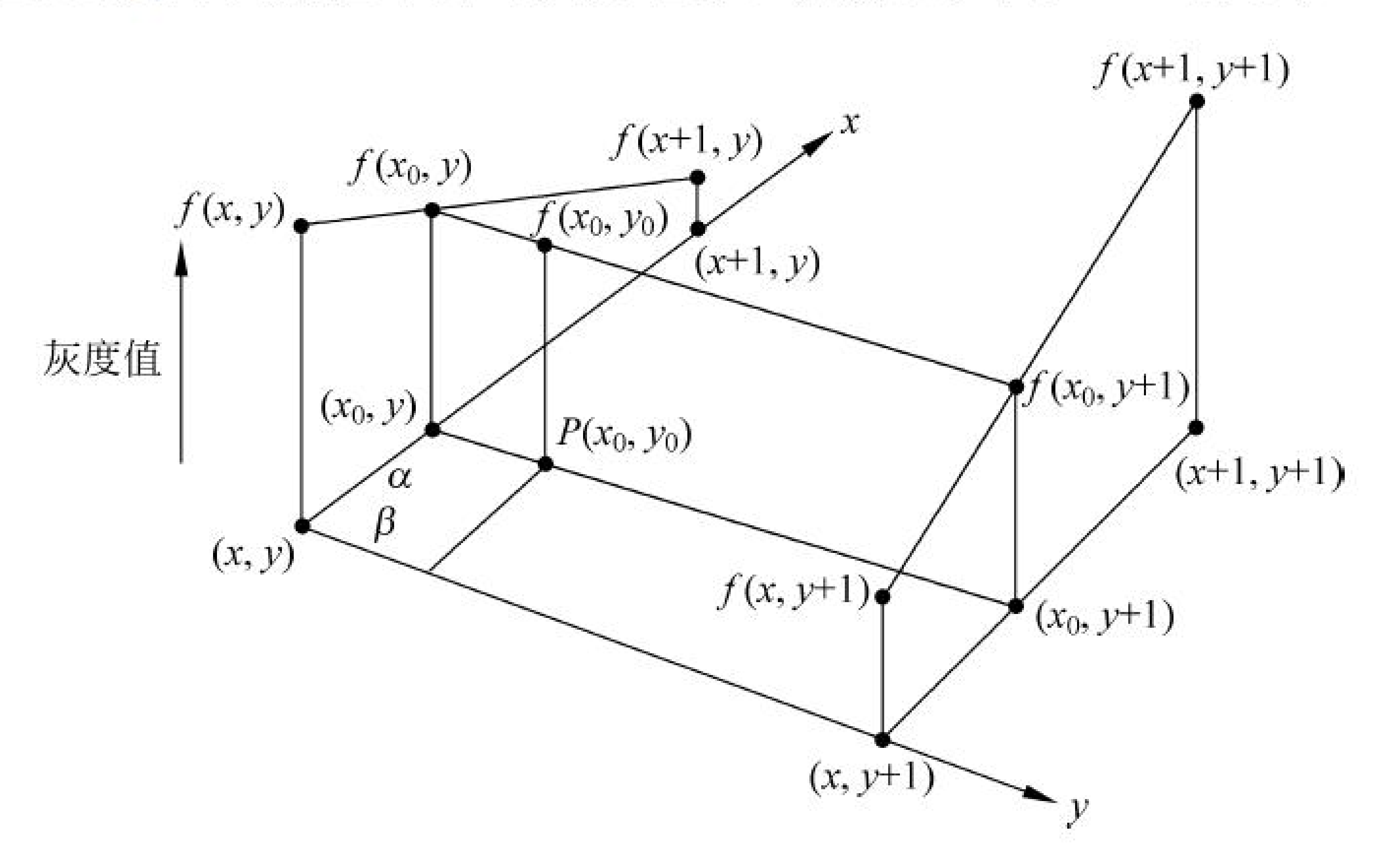


图 4-27　双线性插值

具体计算过程如下：

(1) 计算 α 和 β。

$$\begin{cases}\alpha = x_0 - x \\ \beta = y_0 - y\end{cases} \tag{4-59}$$

(2) 根据 $f(x,y)$ 和 $f(x+1,y)$ 插值求 $f(x_0,y)$。

$$f(x_0,y) = f(x,y) + \alpha[f(x+1,y) - f(x,y)]$$

(3) 根据 $f(x,y+1)$ 和 $f(x+1,y+1)$ 插值求 $f(x_0,y+1)$。

$$f(x_0,y+1) = f(x,y+1) + \alpha[f(x+1,y+1) - f(x,y+1)]$$

(4) 根据 $f(x_0,y)$和 $f(x_0,y+1)$插值求 $f(x_0,y_0)$。

$$\begin{aligned} f(x_0,y_0) &= f(x_0,y)+\beta[f(x_0,y+1)-f(x_0,y)] \\ &= (1-\alpha)(1-\beta)f(x,y)+\alpha(1-\beta)f(x+1,y)+ \\ &\quad (1-\alpha)\beta f(x,y+1)+\alpha\beta f(x+1,y+1) \\ &= f(x,y)+\alpha[f(x+1,y)-f(x,y)]+\beta[f(x,y+1)-f(x,y)]+ \\ &\quad \alpha\beta[f(x+1,y+1)+f(x,y)-f(x,y+1)-f(x+1,y)] \end{aligned} \tag{4-60}$$

式中，$x=[x_0]$，$y=[y_0]$

双线性灰度插值计算方法由于已经考虑到了点 $P(x_0,y_0)$的直接邻点对它的影响，因此一般可以得到令人满意的插值效果。但这种方法具有低通滤波性质，使高频分量受到损失，使图像细节退化而变得轮廓模糊。在某些应用中，双线性灰度插值的斜率不连续还可能会产生一些不期望的结果。

4.5.3 三次内插法

为得到更精确的 $P(x_0,y_0)$点的灰度值，在更高程度上保证几何变换后的图像质量，实现更精确的灰度插值效果，可采用三次内插法等更高阶插值法，如三次样条函数、Legendre中心函数和 $\sin(\pi x)/(\pi x)$函数等，这时既要考虑 $P(x_0,y_0)$点的直接邻点对它的影响，还应考虑到该点周围16个邻点的灰度值对它的影响。

根据连续信号采样定理可知，若对采样值用插值函数 $s(x)=\sin(\pi x)/(\pi x)$进行插值，当采样频率不低于信号谱最高频率的两倍时可以准确地恢复原信号，并可准确地得到采样点间任意点的值。插值函数 $\sin(\pi x)/(\pi x)$特性如图4-28所示。

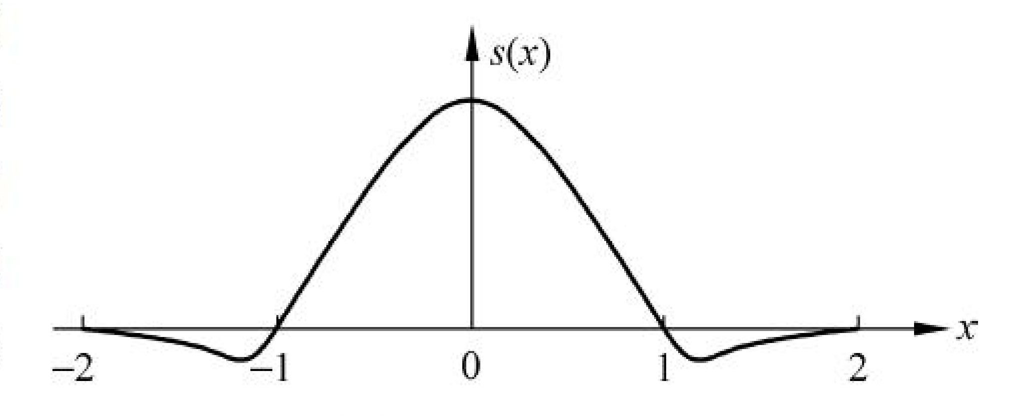


图4-28 插值函数 $\sin(\pi x)/(\pi x)$特性

$s(x)=\dfrac{\sin(\pi x)}{\pi x}$可以采用以下三次多项式近似。

$$s(x)=\begin{cases}1-2|x|^2+|x|^3 & |x|>1 \\ 4-8|x|+5|x|^2-|x|^3 & 2>|x|\geqslant 1 \\ 0 & |x|\geqslant 2\end{cases} \tag{4-61}$$

采用$\dfrac{\sin(\pi x)}{\pi x}$插值函数，可采用下述步骤插值算出 $f(x_0,y_0)$：

(1) 计算 $s(1+\alpha)$、$s(\alpha)$、$s(1-\alpha)$、$s(2-\alpha)$以及 $s(1+\beta)$、$s(\beta)$、$s(1-\beta)$、$s(2-\beta)$。

(2) 根据 $f(x-1,y)$，$f(x,y)$，$f(x+1,y)$，$f(x+2,y)$计算 $f(x_0,y)$。

$$\begin{aligned} f(x_0,y) =& s(1+\alpha)f(x-1,y)+s(\alpha)f(x,y)+s(1-\alpha)f(x+1,y)+ \\ & s(2-\alpha)f(x+2,y) \end{aligned}$$

(3) 按步骤(2)求 $f(x_0,y-1)$，$f(x_0,y+1)$，$f(x_0,y+2)$。

(4) 根据 $f(x_0,y-1)$，$f(x_0,y)$，$f(x_0,y+1)$，$f(x_0,y+2)$计算 $f(x_0,y_0)$。

$$\begin{aligned} f(x_0,y_0) =& s(1+\beta)f(x_0,y-1)+s(\beta)f(x_0,y)+s(1-\beta)f(x_0,y+1)+ \\ & s(2-\beta)f(x_0,y+2) \end{aligned} \tag{4-62}$$

上述计算过程可用矩阵表示为

$$f(x_0, y_0) = \boldsymbol{ABC} \tag{4-63}$$

式中

$$\boldsymbol{A} = [s(1+\alpha) \quad s(\alpha) \quad s(1-\alpha) \quad s(2-\alpha)]$$

$$\boldsymbol{B} = \begin{bmatrix} f(x-1,y-1) & f(x-1,y) & f(x-1,y+1) & f(x-1,y+2) \\ f(x,y-1) & f(x,y) & f(x,y+1) & f(x,y+2) \\ f(x+1,y-1) & f(x+1,y) & f(x+1,y+1) & f(x+1,y+2) \\ f(x+2,y-1) & f(x+2,y) & f(x+2,y+1) & f(x+2,y+2) \end{bmatrix}$$

$$\boldsymbol{C} = [s(1+\beta) \quad s(\beta) \quad s(1-\beta) \quad s(2-\beta)]^{\mathrm{T}}$$

从函数拟合的角度分析，三次内插方法的插值点连线可以实现光滑过渡，插值后的图像灰度一致性较好，三种插值方法中，该方法的插值效果最好，不足之处是计算量增加较多，而且对于图像中细节较多的部分，三次插值具有低通滤波的作用。

习题

1. 数字图像处理算法按功能分为哪几种？

2. 根据图像处理算法的特点，图像在空间域上的处理运算可以分为哪几种类型？各有何特点？

3. 点运算有哪些主要应用领域？为什么数字图像的点运算不会改变图像内各像素的空间位置关系？

4. 为什么点运算又称为灰度变换？灰度变换可以改变图像的对比度吗？为什么？

5. 采用灰度变换能减弱对比度吗？应如何进行运算？

6. 找一幅图像，编写 MATLAB 程序实现如下线性点运算：

$$D_B = aD_A + b$$

其中 $a=1.2$，$b=18$，分析运算结果。

7. 简述图像的点运算具有哪些典型应用。

8. 图像的负片是怎么形成的？应用 MATLAB 在灰度变换程序中具体形成图像负片。

9. 根据代数运算的特点，分析并举例说明代数运算具有哪些典型应用。

10. 为什么梯度图像可以突出表现轮廓？

11. 试编写 MATLAB 程序实现两幅数字图像的相加与相减运算。

12. 几何运算在数字图像处理技术中具有哪些典型应用？

13. 试阐述为什么 2×2 的变换矩阵 $\boldsymbol{T}$ 不能直接实现二维图像的平移变换。

14. 常用的几何运算有哪几种？几何运算在数字图像处理技术中具有哪些典型应用？

15. 什么是齐次坐标？在图像的几何变换中它具有哪些优点？

16. 在图像放大变换中，为什么图像的放大倍数太大时会产生马赛克现象？有哪些方法可以解决这个问题？

17. 编写程序实现图像的镜像变换。

18. 图像的镜像变换包括几种情况？各有何特点？

19. 编写 MATLAB 程序实现图像按任意角度的旋转变换,并分析以下问题:

(1) 根据图像旋转变换公式,旋转变换对图像的质量有无影响?

(2) 若 k 取正整数,观察图像旋转 $k\times 90°$、$k\times 45°$时,会发生什么不同的现象。

20. 什么是图像的复合变换?复合变换可以分为几种情况?

21. 试简述如何应用控制点法进行图像的几何校正。

22. 根据双线性插值原理,分析为何双线性插值具有低通滤波作用。

第 5 章 图像空域增强

CHAPTER 5

图像在获取、传输、使用过程中，可能会产生噪声或使图像质量降低，图像增强即有针对性地消除和抑制图像噪声、提高图像质量。简而言之，图像增强是根据应用需要突出图像中的某些“有用”信息，削弱或去除不需要的信息，以扩大图像中不同物体特征之间的差别。处理的目的是使处理后的图像对于特定应用而言，比原始图像更合适，或者使图像特征信息提取更方便，为进一步对图像进行分析奠定基础。由于具体应用的目的和要求不同，因而“有用”的含义和标准也不尽相同。一般情况下，经过增强处理后，图像的视觉效果会发生变化，这种变化意味着某些特定信息得到了增强，或者图像的视觉效果得到了改善。

图像增强在航空航天、工业自动化生产、生物医学、社会公共事务管理等方面得到了广泛的应用。比如机械零部件的检查和识别、印刷电路板的检查、食品包装出厂前的质量检查、X 射线成像、工件尺寸测量、集成芯片内部电路缺陷检测等。需要注意的是，图像增强算法并不能增加原始图像的信息，而是通过某种技术手段有选择地突出对某一具体应用有价值的信息。即图像增强只通过突出某些信息以增强对这些信息的辨识能力，而其他信息则被压缩。也就是说，图像的增强处理并不是一种无损处理。例如，图像平滑处理算法中经常采用低通滤波法，虽然消除了图像的噪声，但图像的空间纹理特征却被削弱，图像从整体上显得比较模糊。因此，图像噪声的消除是以纹理信息（高频信息）的减弱为代价而实现的。本章围绕图像增强技术的原理与方法，首先介绍数字图像噪声及其产生的原因、图像增强处理方法分类，然后介绍图像直方图增强法、代数增强法、空域线性平滑滤波、空域非线性滤波和锐化滤波等增强方法。

5.1 图像噪声

对于数字图像处理而言，噪声是指图像中的非本源信息。图像中各种妨碍人们对其信息接受的因素即可称为图像噪声。在理论上，噪声可以定义为不可预测，只能用概率统计方法来认识的随机误差。因此，噪声会影响人的感官对所接收的源信息的准确理解。在理论上，噪声只能通过概率统计的方法来认识和研究。从一般情况分析，图像噪声可认为是多维随机信号，可以采用概率分布函数、概率密度函数，以及均值、方差、相关函数等对噪声进行描述和分析。

5.1.1 图像噪声的产生

目前，大多数数字图像系统中，输入图像都是通过扫描方式将多维图像变成一维电信号，再对其进行存储、处理和传输等，最后形成多维图像信号。这一系列复杂过程中，图像数字化设备、电气系统和外界影响将使得图像噪声的产生不可避免。例如，处理高放大倍数遥感图片的X射线图像系统中的噪声去除等已成为不可或缺的技术。

5.1.2 图像噪声的分类

图像噪声按其产生的原因可分为外部噪声和内部噪声。外部噪声指系统外部干扰以电磁波或经电源串进系统内部而引起的噪声，如电气设备、天体放电现象等引起的噪声。内部噪声指成像系统本身原因引起的噪声。一般情况下，数字图像中常见的内部干扰主要包括如下几种：

（1）由光和电的基本性质所引起的噪声。如电流的产生是因电子或空穴粒子的集合和定向运动而形成，因这些粒子的随机性运动会产生散粒噪声；而导体中自由电子的无规则热运动会产生热噪声；根据光的粒子性，图像是由光量子所传输，而光量子密度随时间和空间变化也可形成光量子噪声等。

（2）设备元器件及材料本身引起的噪声，如磁带、磁盘表面缺陷所产生的噪声。

（3）系统内部设备电路所引起的噪声，包括电源系统引入的交流噪声、偏转系统和钳位电路引起的噪声等。

（4）电器部件机械运动产生的噪声，如数字化设备的各种接头因抖动引起的电流变化所产生的噪声，磁头、磁带抖动引起的抖动噪声等。

需要指出的是，噪声分类并不是绝对的，按不同的性质有不同的分类方法。例如，依据统计特征，图像噪声可分为平稳噪声和非平稳噪声两种，其中统计特性不随时间变化的噪声称为平稳噪声，统计特性随时间变化的噪声称为非平稳噪声。根据噪声与信号之间的关系，可分为加性噪声和乘性噪声。加性随机噪声方法成熟，而乘性随机噪声处理方法目前还没有成熟和通用的处理方法。按噪声频谱分类，又可以分为白噪声、$1/f$ 噪声和三角噪声。频谱均匀分布的噪声称为白噪声；频谱与频率成反比的称为 $1/f$ 噪声；而与频率平方成正比的称为三角噪声等。

5.1.3 图像噪声的特点

如图5-1所示是一幅含有噪声的图像。一般情况下，图像中的噪声有以下三个特点。



图5-1 含有噪声的图像

1. 随机性

由于噪声在图像中是随机出现的，所以其分布和大小是随机的，对于噪声的处理，需根据具体噪声的特点，采用概率统计方法进行分析和处理。

2. 叠加性

根据噪声与信号之间的关系，噪声包括加性噪声和乘性噪声。理论上，加性随机噪声方法成熟，且处理比较方

便；而乘性随机噪声处理方法目前还没有成熟的理论，并且处理起来非常复杂。一般条件下，现实生活中所遇到的绝大多数图像噪声均可以认为是加性噪声。在图像的串联传输系统中，各个串联部件引起的噪声一般具有叠加效应，导致信噪比下降。

3. 噪声与图像之间具有相关性

通常情况下，摄像机的信号和噪声相关，明亮部分噪声小，黑暗部分噪声大。数字图像处理技术中存在的量化噪声与图像相位相关。例如，图像内容接近平坦时，量化噪声呈现伪轮廓，但此时图像信号中的随机噪声会因为颤噪效应反而使量化噪声变得不很明显。

改善被噪声污染的图像质量有两种方法：一是不考虑图像噪声的原因，只对图像中某些部分加以处理或突出有用的图像特征信息，改善后的图像并不一定与原图像信息完全一致。这一类改善图像特征的方法就是图像增强技术，主要目的是要提高图像的可辨识性。另一类方法是针对图像产生噪声的具体原因，采取技术方法补偿噪声影响，使改善后的图像尽可能地接近原始图像。这类方法称为图像恢复或图像复原技术。

5.2 图像增强处理分类

图像增强处理方法根据处理过程所在的空间不同，可分为基于空间域的增强方法和基于频率域的增强方法两大类，如图 5-2 所示。

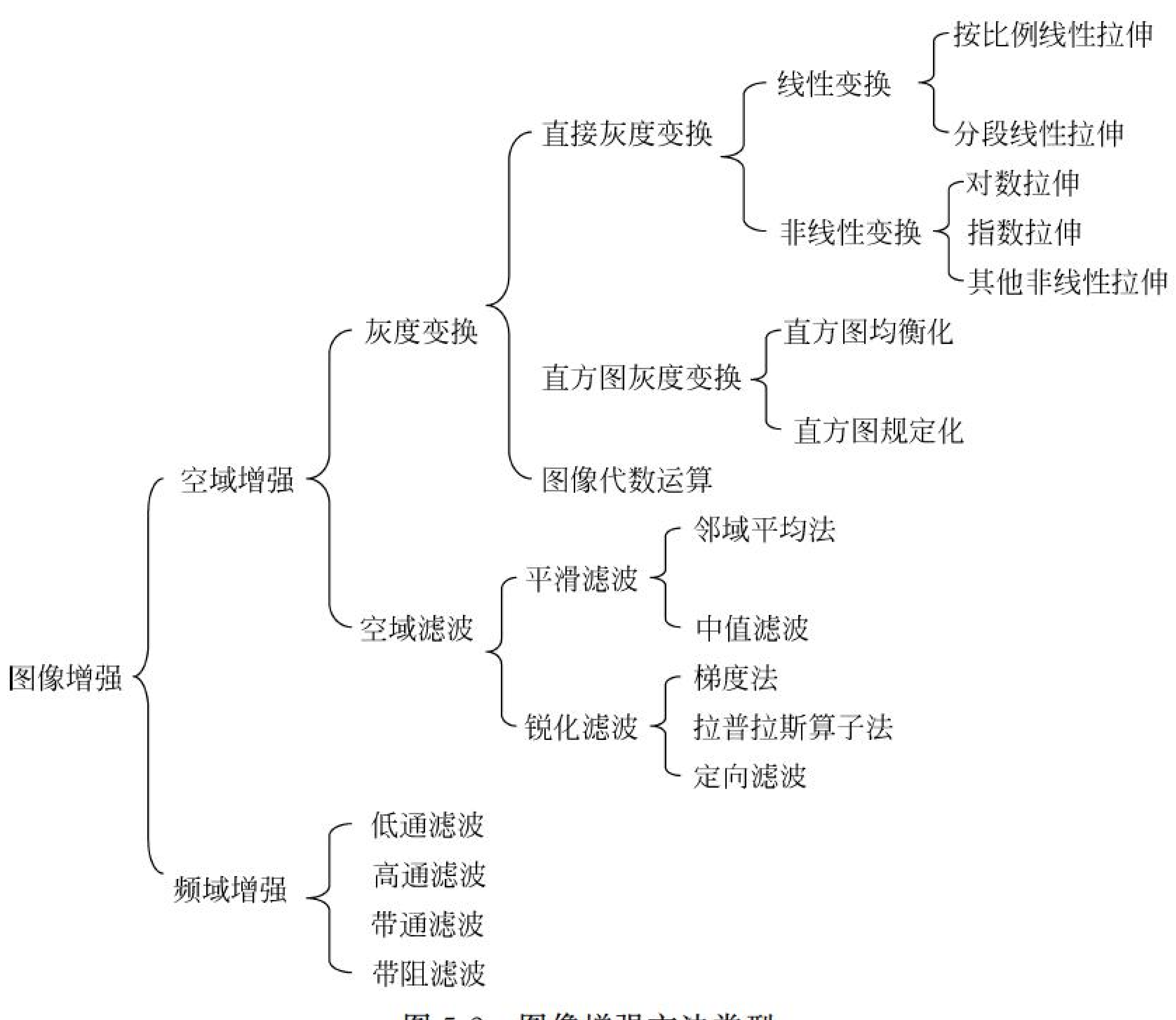


图 5-2 图像增强方法类型

此外，图像增强技术按所处理对象的不同还可分为灰度图像增强和彩色图像增强；按增强的目的还可分为光谱信息增强、空间纹理信息增强和时间信息增强。通常情况下，如果没有特别说明，一般均指对灰度图像的增强。

5.2.1 空域增强法

基于空域的增强方法直接在图像所在的二维空间进行处理，即直接对每一像素点的灰度值进行处理。根据所采用的技术不同又可分为灰度变换和空域滤波两类方法。

灰度变换是基于点操作的增强方法，将每一像素点的灰度值按照一定的数学变换转换为一个新的灰度值。基于灰度变换原理的图像增强方法非常丰富，如增强处理中常用的直接灰度变换（包括线性拉伸和非线性拉伸）、对比度增强、直方图均衡化、直方图规定化和图像的代数运算等方法都属于灰度变换技术。

空域滤波是基于邻域处理的增强方法，它应用某一模板对每个像素点与其周围邻域的所有像素点进行某种确定数学运算得到该像素点新的灰度值，输出值的大小不仅与该像素点的灰度值有关，而且还与其邻域内的像素点的灰度值有关。常用的图像平滑滤波与锐化滤波技术就属于空域滤波的范畴。

5.2.2 频域增强法

频域增强法首先将图像从空域按照某种变换模型（如傅里叶变换或其他变换等）变换到频域，然后在频域对图像进行处理，再将其反变换到空域。通常包括低通、高通、带通和带阻四种典型的滤波器结构。

5.2.3 图像增强效果评价

目前对图像增强效果的评价主要包括定性评价和定量评价两个方面。

（1）定性评价主要根据人的主观感觉，对图像增强的视觉效果进行评判，一般主要对图像的清晰度、色调、纹理等几方面进行主观评价。定性分析的不足是与评价者的主观性密切相关，即对同一幅被增强的图像，不同的人可能有不同的评价。定性分析的主要优点是可以从一幅图像中有选择地对具体研究对象进行重点比较和评价，即定性分析可以对图像的局部或具体研究目标进行评价，具有灵活性和广泛的适应性。

（2）定量评价图像增强效果目前还没有业界统一接收的标准与尺度，目前通常采用的方法是从图像的信息量、标准差、均值、纹理度量值和具体研究对象的光谱特征值等方面与原始图像进行比较评价。定量分析的最大优点是客观公正，但通常是对一幅图像从整体上进行统计分析，很难对图像的局部或具体对象进行评价，而图像整体的定量分析容易受到噪声等因素的影响。因此，对图像增强效果的评价一般以定性分析为主。

需要强调的是，评价一个图像增强算法的性能优越与否是比较复杂的，增强效果的好坏不仅与具体算法有一定的关系，还与原始图像的数据特征直接相关。一个对图像 A 效果好的增强算法不一定适合于图像 B。因此，为了得到满意的图像增强效果，一般情况下应同时比较几种增强算法，从中选出既视觉效果好、计算量小，又满足要求的最优算法。

5.3 直接灰度变换增强

顾名思义，直接灰度变换是直接对数字图像的像素点进行数学变换的一种图像增强方法。它是空域灰度变换（一般简称为灰度变换）增强法中一类具有广泛应用的方法。灰度变

换增强法是通过寻求数学变换函数使图像的对比度提高，在图像空间所进行的灰度变换本质上是一种点处理方法。根据处理的过程，常用空域灰度变换包括以下三种：

（1）直接灰度变换，通过选择合适的变换公式依次对每个像素点进行处理。

（2）基于灰度直方图的灰度变换。

（3）对多个灰度图像进行代数运算实现灰度变换。

直接灰度变换增强法包括基于数学函数的灰度变换方法和基于传统光学增强方法两类。基于数学函数的灰度变换方法包括线性拉伸变换、非线性拉伸变换方法。基于传统光学的图像增强方法主要包括亮度调节、对比度调节、图像反转等多种方法。实际上，传统的光学增强方法也可通过数学变换实现。

5.3.1　线性灰度拉伸

线性灰度拉伸（又称为线性拉伸）是将输入图像的灰度值的动态范围按线性关系拉伸扩展至指定范围或灰度的整个动态范围，图 5-3 所示是一个常用的灰度线性拉伸函数关系。对于常见的 8 位灰度图像而言，其最大动态范围是[0,255]。

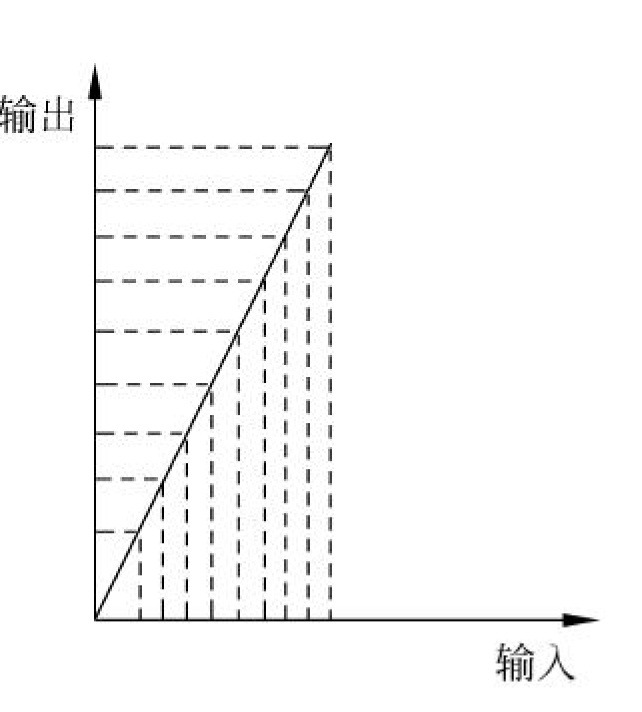


图 5-3　线性变换对灰度的拉伸效果

线性拉伸可提高图像的质量，如例 5-1 所示。

例 5-1：灰度图像线性拉伸效果

如图 5-4 所示为线性拉伸增强图像视觉效果示例。图 5-4(a)为 128 灰度级图像，图 5-4(b)为经灰度线性拉伸为 256 灰度级图像，灰度拉伸以后，图像在亮度和对比度等方面具有明显的改善效果。

设 $f(x,y)$表示输入图像灰度值，$g(x,y)$表示输出图像灰度值，线性灰度拉伸采用的变换公式一般为如下线性方程：



(a) 128灰度级



(b) 256灰度级

图 5-4　线性灰度变换增强图像的效果

$$g(x,y) = a \times f(x,y) + b \tag{5-1}$$

其中，a、b 的值需根据具体应用需求确定。线性拉伸还可分为按比例线性拉伸和分段线性拉伸两种方法。

1. 按比例线性拉伸

如图 5-5 所示，按比例线性拉伸包括图 5-5(a)所示的均匀线性拉伸和图 5-5(b)所示的限幅线性拉伸两种情况。

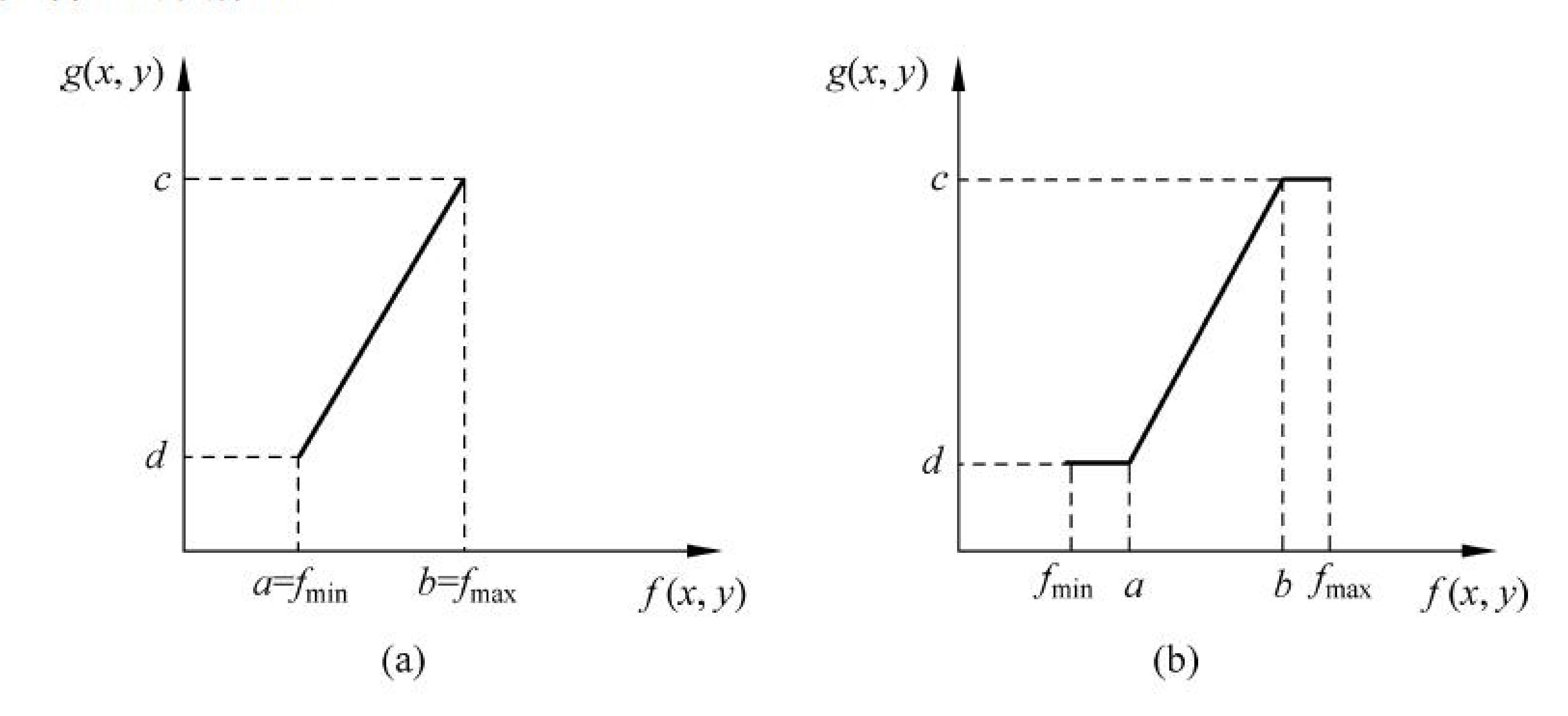


图 5-5 按比例线性拉伸

1）均匀线性拉伸

均匀线性拉伸是线性拉伸中最简单、最常用的一种方法，根据图 5-5(a)可得出其灰度变换关系为

$$g(x,y)=\frac{g_{\max}-g_{\min}}{f_{\max}-f_{\min}}[f(x,y)-f_{\min}]+g_{\min} \tag{5-2}$$

即

$$g(x,y)=\frac{c-d}{b-a}[f(x,y)-a]+d \tag{5-3}$$

2）限幅线性拉伸

限幅线性拉伸是一种在某些特定应用中具有重要意义的线性拉伸方法，它将原始图像的低灰度值和高灰度值像素点的灰度级进行了适当的归并，而对中间部分的灰度值进行了拉伸。实际图像处理中，根据图像处理的不同应用需求，原始图像的归并部分和拉伸部分可以灵活选取。根据图 5-5(b)，其灰度变换公式为

$$g(x,y)=\begin{cases} c & f_{\min}\leqslant f(x,y)<a \\ \dfrac{f(x,y)-a}{b-a}(c-d)+d & a\leqslant f(x,y)<b \\ d & b\leqslant f(x,y)\leqslant f_{\max} \end{cases} \tag{5-4}$$

2. 分段线性拉伸

在图像处理实践中，为了突出图像中所关注的研究对象，常常要求分段按不同比例拉伸，如图 5-6 所示，即局部拉伸某一范围的灰度值，或对不同范围的灰度值进行不同的拉伸处理，即分段线性拉伸。

实际上，分段线性拉伸的结果是仅将某一局部范围的灰度值进行了拉伸，而其余部分的灰度值是被压缩了。对应于图 5-6 所示的两种分段线性拉伸的变换公式如下：

(1) 对于图 5-6(a)，其变换公式为

$$g(x,y)=\begin{cases} \dfrac{c}{a}f(x,y) & 0\leqslant f(x,y)<a \\ \dfrac{M_g-c}{M_f-a}(f(x,y)-a)+c & a\leqslant f(x,y)\leqslant M_f \end{cases} \tag{5-5}$$

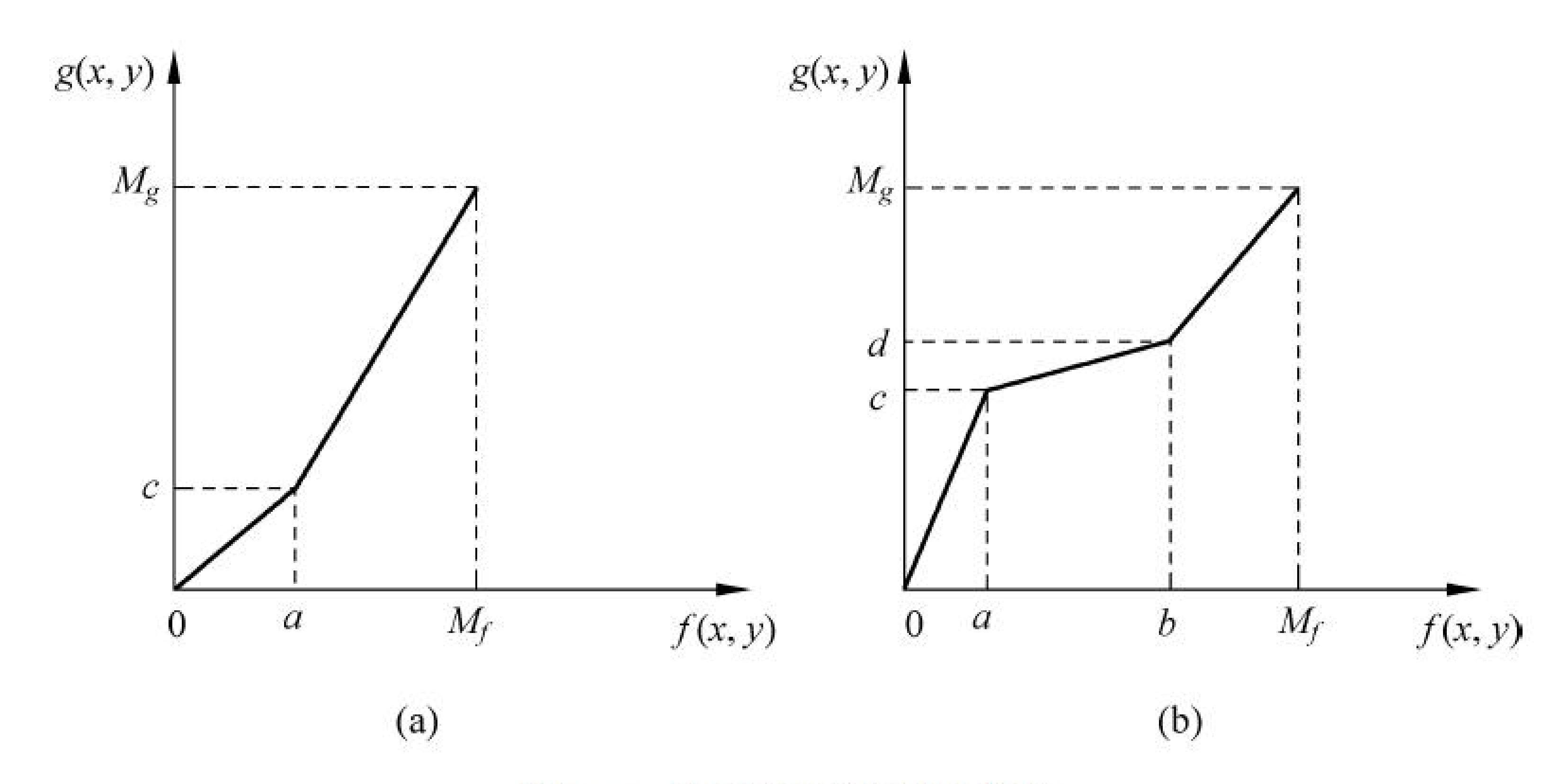


图 5-6　分段(局部)线性拉伸

(2) 对于图 5-6(b),其变换公式为

$$g(x,y)=\begin{cases}\dfrac{c}{a}f(x,y) & 0\leqslant f(x,y)<a\\ \dfrac{d-c}{b-a}(f(x,y)-a)+c & a\leqslant f(x,y)<b\\ \dfrac{M_g-d}{M_f-b}(f(x,y)-b)+d & b\leqslant f(x,y)\leqslant M_f\end{cases}\tag{5-6}$$

5.3.2　非线性拉伸

非线性拉伸是指采用非线性变换函数对输入图像的灰度进行拉伸变换。它与分段线性拉伸的相同之处是有选择地对某一灰度值范围进行了拉伸,即不是对图像的整个灰度值范围进行拉伸,图像灰度值的某些特定范围被拉伸了,而其他灰度值范围则有可能被压缩;不同点是,非线性拉伸不是通过在不同灰度值区间选择不同的线性函数来实现分段拉伸或压缩,非线性拉伸在整个灰度值范围内采用统一的非线性变换函数,利用非线性变换的性质实现对不同灰度值区间的拉伸与压缩。常用的非线性拉伸函数包括对数拉伸、指数拉伸、平方函数拉伸、阈值函数拉伸和窗口函数拉伸等多种处理方法。

1. 对数拉伸

对数拉伸的形式为

$$g(x,y)=\log_a[f(x,y)]\tag{5-7}$$

式中,a 为对数的底数,可以根据需要灵活选择。在实际应用中,一般取 $a=\mathrm{e}$,即选择自然对数,具体形式为

$$g(x,y)=C\ln[f(x,y)+1]\tag{5-8}$$

式中,C 为尺度比例系数,用于调节动态范围,$[f(x,y)+1]$是为了避免对零求对数。如图 5-7 所示,对数扩展可以将图像的低亮度区进行大幅拉伸,但是高亮度区却被压缩。

2. 指数拉伸

指数拉伸的变换形式为

$$g(x,y)=b^{C[f(x,y)-a]}\tag{5-9}$$

式中,参数 a 可以改变曲线的起始位置;参数 C 可以改变曲线的变化速率,其变换函数曲线

如图 5-8 所示，指数扩展可以对图像的高亮度区进行大幅扩展。

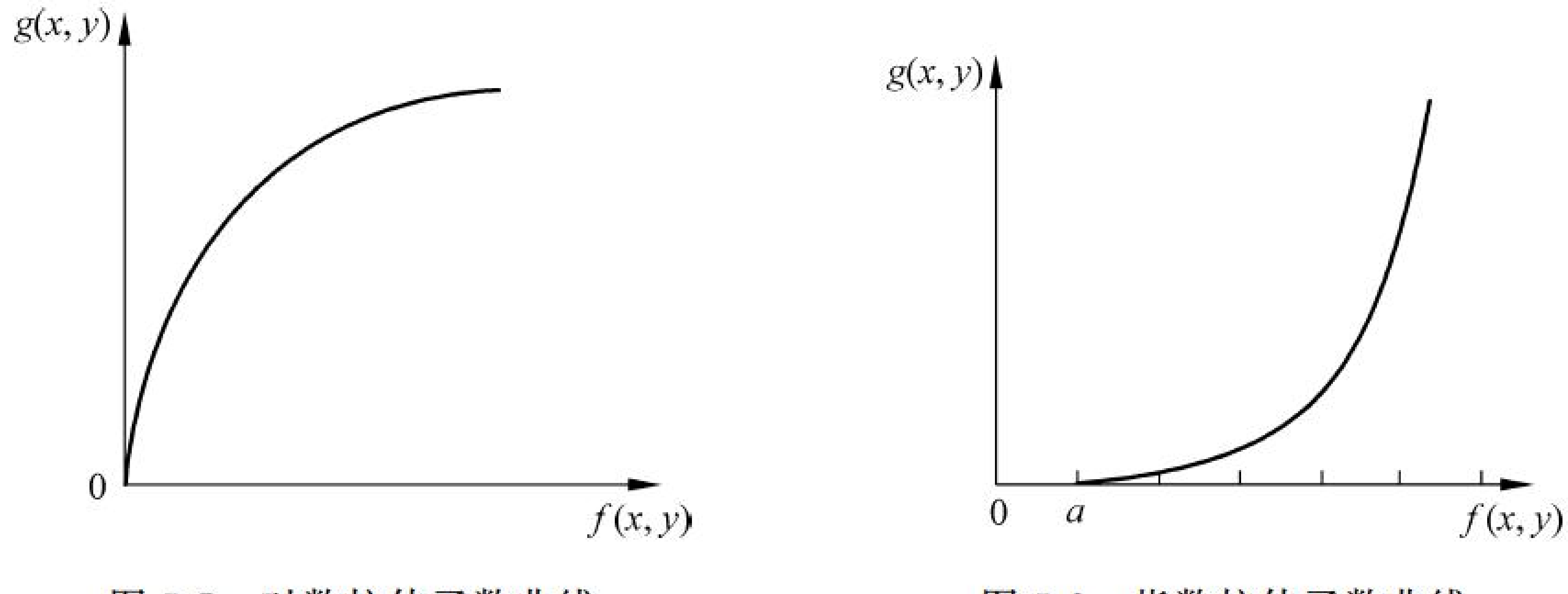


图 5-7　对数拉伸函数曲线　　　　图 5-8　指数拉伸函数曲线

3. 其他拉伸函数

除以上两种典型的非线性拉伸函数之外，其他常用非线性拉伸还包括平方函数、阈值函数、窗口函数等多种形式。

5.3.3　光学增强

图像的光学增强法是一种非常传统的方法，许多光学增强方法曾经一度在模拟图像处理中得到较广泛应用。如图像的反转很早就已经在照相技术中应用，普通黑白底片和照片的关系就是一个相互反转的关系，所谓图像反转就是使图像的灰度值反转。包括亮度调整、对比度调节等在内的传统增强方法也可以通过数学变换函数来实现。

5.4　基于直方图的图像增强

基于灰度直方图的图像增强方法是三类灰度变换方法中的第二类方法。直方图是对图像每一灰度值出现的频率进行统计的一种直观表示方法。基于直方图的图像增强方法是以概率统计学理论为基础的，如图像的明暗状况和对比度等特征信息都可以通过直方图反映出来。如图 5-9 所示，这是同一幅图像经过不同处理后的直方图的四种分布情况，在图 5-9(a)中，大部分像素点的灰度值小于 50，表示该图像亮度偏低；同样经过分析可以得出，图 5-9(b)亮度偏高，图 5-9(c)所示图像灰度值范围偏小，而图 5-9(d)所示图像灰度值范围比较正常。根据该图可以看出，一幅图像的直方图从整体上描述了一幅图像的概貌特征，因此，可以通过修改直方图的方法来调整一幅数字图像的灰度分布。常用的方法有直方图均衡化和直方图规定化等。

5.4.1　基本原理

一幅给定的图像(原始图像)的灰度级通过归一化处理可以认为分布在 $0 \leqslant r \leqslant 1$ 的范围内，对[0,1]区间内的任一灰度级 r 可以进行如下形式变换：

$$s = T(r) \tag{5-10}$$

根据式(5-10)的变换，原始图像每一个像素点的灰度值 r 都有一个变换灰度值 s 对应。假定式(5-10)表示的变换函数满足以下两个条件：

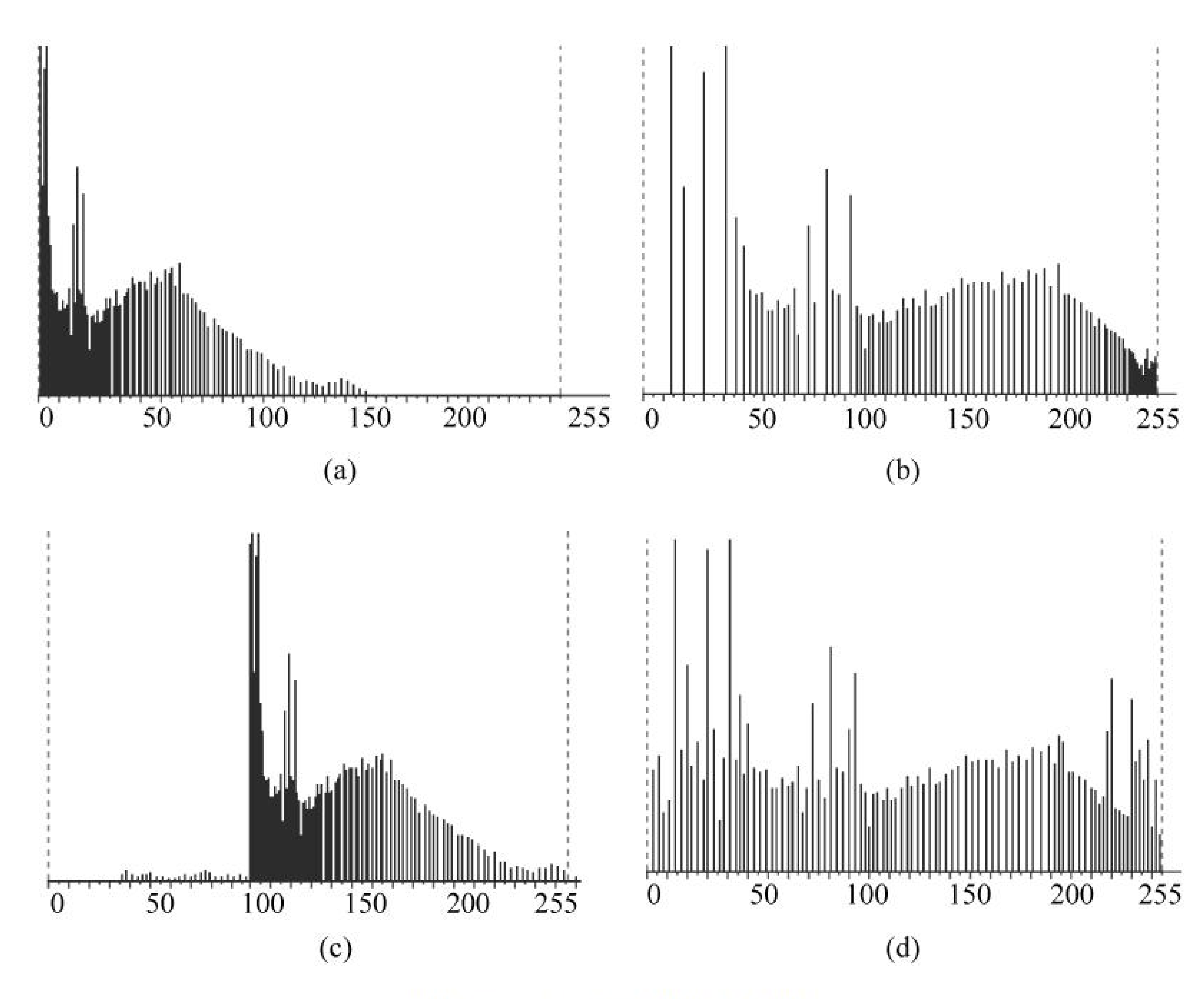


图 5-9 不同类型的直方图

(1) 在 $0 \leqslant r \leqslant 1$ 区间内，$s = T(r)$ 值为单调增加。

(2) 对于 $0 \leqslant r \leqslant 1$，有 $0 \leqslant s = T(r) \leqslant 1$。

上述条件(1)用于保证原始图像各灰度级在变换后仍保持从黑到白(或从白到黑)的单调排列顺序，条件(2)用于在理论上严格保证变换之后，灰度值范围的一致性。根据数学函数关系，从 s 到 r 的逆变换用下式表示：

$$r = T^{-1}(s), \quad 0 \leqslant s \leqslant 1 \tag{5-11}$$

由于变换关系 T 满足上述两个条件，根据数学分析，$T^{-1}(s)$ 也满足这两个条件。

由概率论知识可知，如果已知随机变量 r 的概率密度为 $P_r(r)$，而随机变量 s 是 r 的函数，则 s 的概率密度 $P_s(s)$ 可以由 $P_r(r)$ 求出，主要过程如下：

(1) 根据分布函数的定义可以求得随机变量 s 的分布函数 $F_s(s)$，即

$$F_s(s) = \int_{-\infty}^{r} P_r(x)\,\mathrm{d}x$$

(2) 根据分布函数与概率密度函数之导数关系，可得出 $P_s(s)$，即

$$P_s(s) = P_r(r)\frac{\mathrm{d}r}{\mathrm{d}s}\Big|_{r=T^{-1}(s)} = P_r(r)\frac{\mathrm{d}[T^{-1}(s)]}{\mathrm{d}s} \tag{5-12}$$

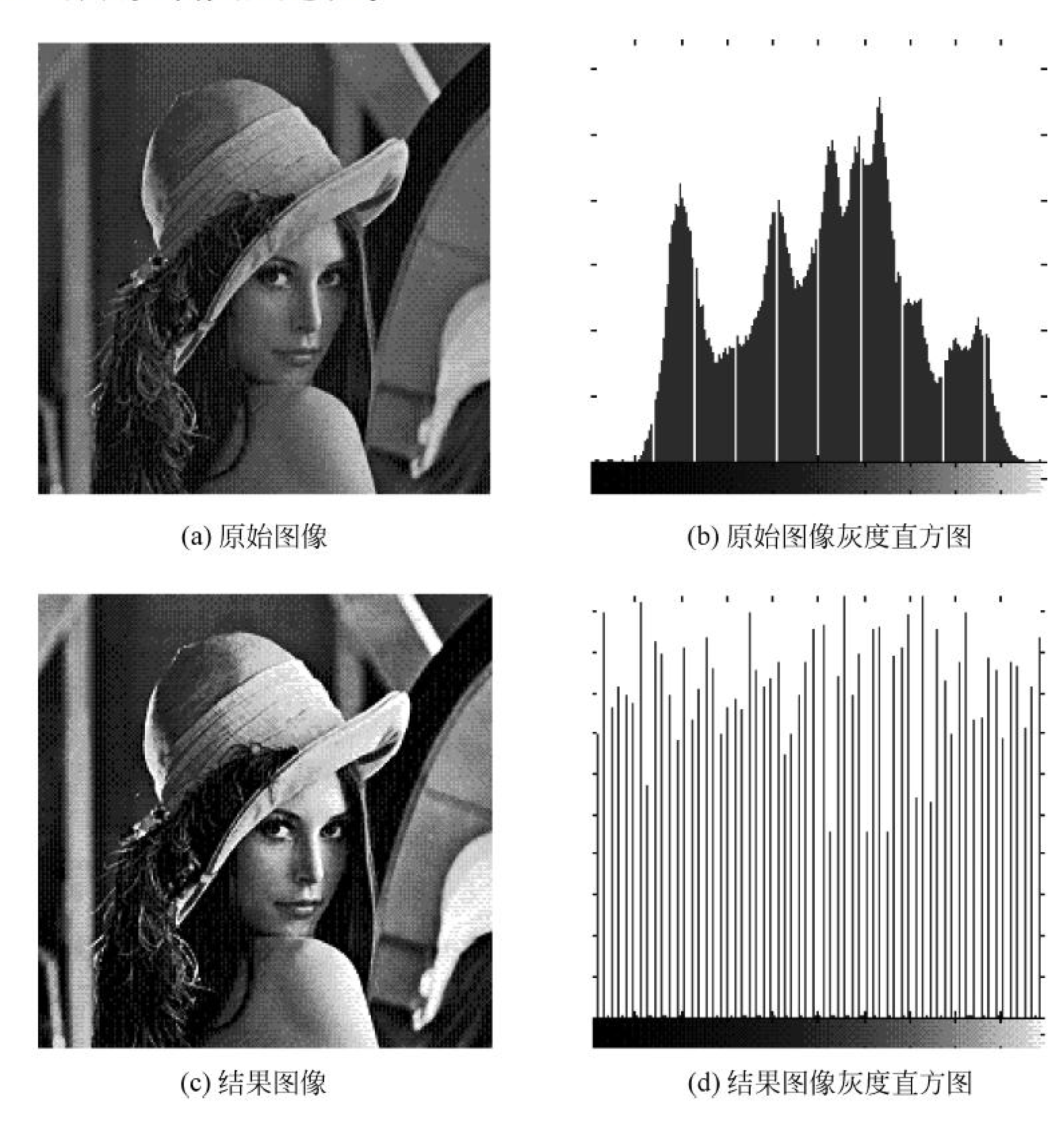
综上所述，直方图增强的技术基础是通过 s 与 r 之间的变换函数 $T(r)$ 控制图像灰度级的概率密度函数，从而改善图像的灰度分布情况。

通常情况下，直方图均衡化对于改善图像对比度和亮度具有明显的效果，如例 5-2 所示。

例 5-2：直方图均衡化对图像效果的改善

本示例给出了一幅图像直方图均衡化前后的对比效果。图 5-10(a)是原始图像、

图 5-10(b)为原始图像的直方图。由于原始图像的灰度值分布在中间灰度值一个较窄的区间,而且亮度偏低,因而图像显得非常模糊。经过直方图均衡化变换后,如图 5-10(c)和图 5-10(d)所示,图像的灰度值动态范围增加,图像的亮度得到了一定程度的提升,图像从整体上给人一种明亮、清晰的感觉。



(a) 原始图像　(b) 原始图像灰度直方图

(c) 结果图像　(d) 结果图像灰度直方图

图 5-10　直方图均衡化效果

5.4.2　直方图均衡化

直方图均衡化处理是以累积分布函数为基础的直方图修改法。直方图均衡化的目的是将原始图像的直方图变为均衡分布的形式,将一非均匀灰度概率密度分布图像,通过寻求某种灰度变换,变成一幅具有均匀概率密度分布的目的图像。

若一幅图像的直方图是均衡的,其图像灰度级的取值范围为 L,则其概率密度函数为

$$P_s(s) = \frac{1}{L}$$

由于灰度级已经过归一化处理,所以有

$$\begin{cases} L = 1 \\ P_s(s) = 1 \end{cases}$$

故式(5-12)可改写为

$$\mathrm{d}s = P_r(r)\mathrm{d}r$$

对上式进行积分，可得

$$s=\int_0^r P_r(r)\mathrm{d}r$$

如果取变换函数，则有

$$\int_0^r P_r(r)\mathrm{d}r = T(r) \tag{5-13}$$

由此可知，以 r 的累积分布函数作为变换函数，就可以拓展原始图像灰度值的动态范围，因而可以将概率密度函数为 $P_r(r)$的图像变换为具有均匀概率密度函数 $P_s(s)$的图像。变换式(5-13)是原始图像的累积概率密度函数，并且满足直方图变换函数的两个条件。

上述推导是以连续变量为基础的，若灰度值是离散值，则可以采用灰度频数近似代替概率值。设图像有 L 个灰度级，图像像素点总数为 n，n_k代表灰度级为 r_k的像素点的数量，则第 k 个灰度级出现的概率可表示为

$$P_r(r_k)=\frac{n_k}{n}\quad 0\leqslant r_k\leqslant 1,\ 0\leqslant k\leqslant L-1$$

对于离散情况，式(5-13)为

$$s_k=T(r_k)=\sum_{j=0}^{k}P_r(r_j)\sum_{j=0}^{k}\frac{n_j}{n}\quad 0\leqslant r_j\leqslant 1,\ 0\leqslant k\leqslant L-1 \tag{5-14}$$

对应的逆变换为

$$r_k=T^{-1}(s_k)$$

下面通过一个具体实例来说明如何对一幅图像进行直方图均衡化处理以及图像经过直方图均衡化后的增强效果。

例 5-3：直方图均衡化计算原理与过程

现有一幅灰度数字图像，像素数为 64×64，共有 8 个灰度级，其灰度级分布如表 5-1 所示，现要求对其进行均衡化处理。原始图像的灰度直方图如图 5-11(a)所示，图中横坐标轴标记 0、1、…、7 分别代表 0、1/7、…、1。

表 5-1　64×64 大小图像灰度分布表

r_k	n_k	$P_r(r_k)=\frac{n_k}{n}$
$r_0=0$	790	0.19
$r_1=1/7$	1023	0.25
$r_2=2/7$	850	0.21
$r_3=3/7$	656	0.16
$r_4=4/7$	329	0.08
$r_5=5/7$	245	0.06
$r_6=6/7$	122	0.03
$r_7=1$	81	0.02

直方图均衡化的处理过程如下：

(1) 求变换函数。

根据变换函数式(5-14)，计算各灰度级 s_k，即

$$s_0=T(r_0)=\sum_{j=0}^{0}P_r(r_j)=0.19$$

$$s_1 = T(r_1) = \sum_{j=0}^{1} P_r(r_j) = 0.19 + 0.25 = 0.44$$

$$s_2 = T(r_2) = \sum_{j=0}^{2} P_r(r_j) = 0.19 + 0.25 + 0.21 = 0.65$$

依次计算可分别得出：$s_3=0.81$，$s_4=0.89$，$s_5=0.95$，$s_6=0.98$，$s_7=1$。

根据以上所求出的 s_k 和 r_k 之间的离散值关系，可以得出 s_k 和 r_k 之间的离散函数关系(即变换函数)如图 5-11(b)所示。

(2) 计算均衡化之后的灰度级。

由于原图像的灰度级只有 8 级，变换之后的 s_k 只能选择最接近的一个灰度级，因此需对 s_k 进行舍入处理，各 s_k 将以 1/7 为量化单位进行舍入运算，结果如下：

$$s_0 \approx 1/7, s_1 \approx 3/7, s_2 \approx 5/7, s_3 \approx 6/7, s_4 \approx 6/7, s_5 \approx 1, s_6 \approx 1, s_7 \approx 1。$$

根据对 s_k 的舍入结果，均衡化后的灰度级仅有 5 个级别，分别是：

$$s_0 = 1/7, s_1 = 3/7, s_2 = 5/7, s_3 = 6/7, s_4 = 6/7$$

(3) 计算对应每个 s_k 的像素点数量。

因为 $r_0=0$，经映射到 $s_0=1/7$，所以有 790 个像素点取 s_0 这个灰度值；$r_1=1/7$ 映射到 $s_1=3/7$，所以有 1023 个像素点取值 $s_1=3/7$ 这个灰度值，同理有 850 个像素点取值 $s_2=5/7$ 这个灰度值，由于 r_3 和 r_4 都映射到 $s_3=6/7$ 灰度值，所以有 656+329=985 个像素点取此灰度值，同样有 245+122+81=448 个像素点取 $s_4=1$ 的灰度值。

(4) 绘出均衡化以后的直方图。

对于均衡以后的每一灰度级，计算各 n_k/n 即可得到均衡化后的直方图，如图 5-11(c)所示。

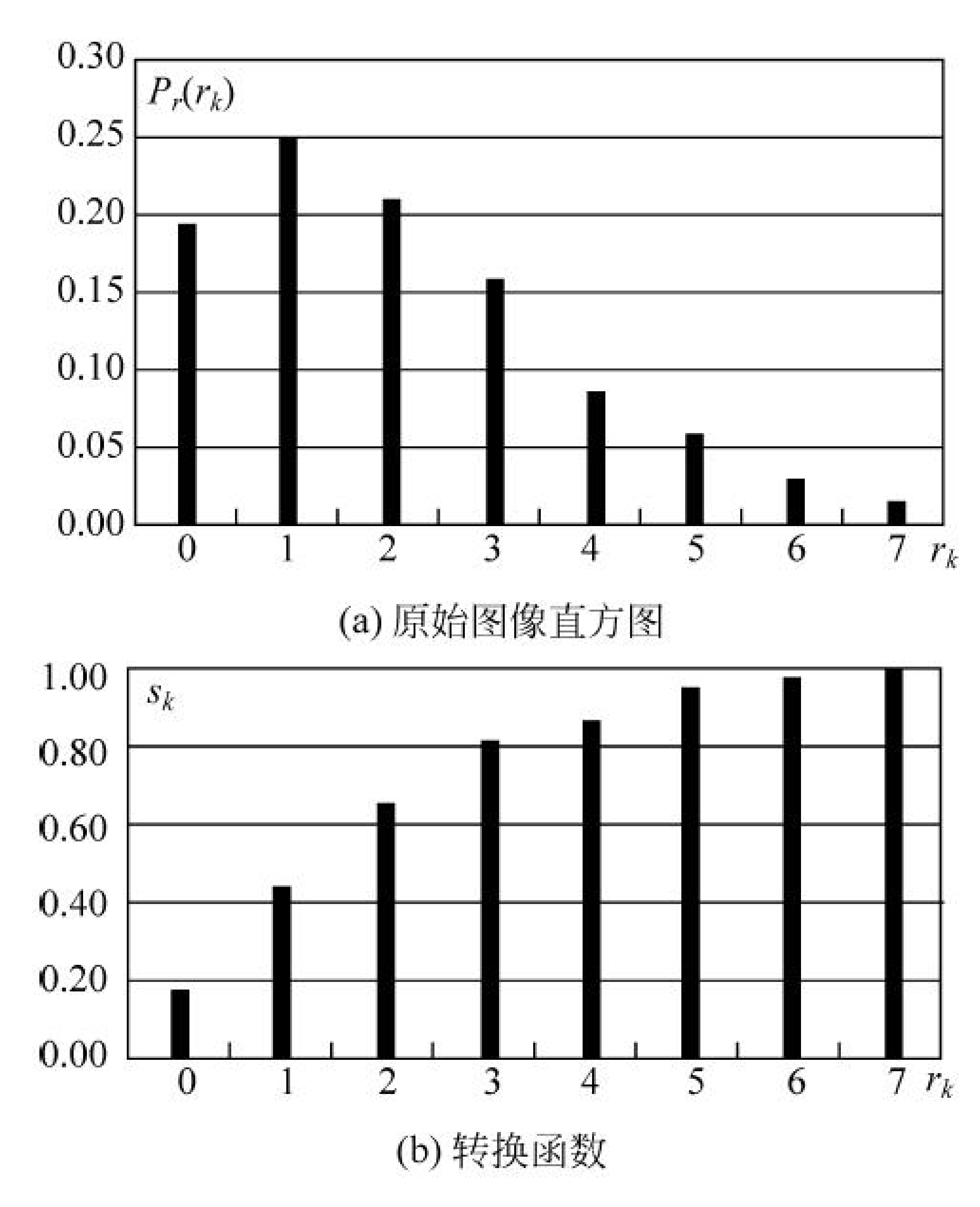
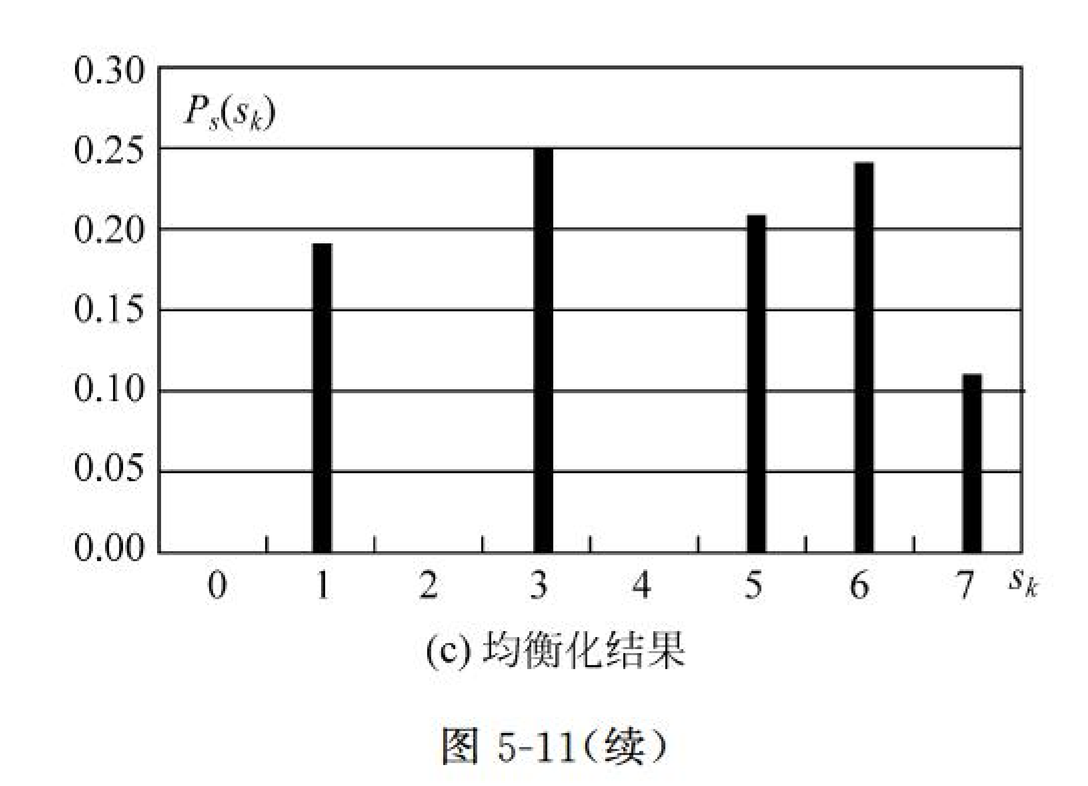


图 5-11 直方图均衡化过程



(c) 均衡化结果

图 5-11(续)

根据本例可以看出，对于连续图像，理论上可以完全实现均衡化，但在离散灰度级情况下，利用累积分布函数作为灰度变换函数，所得到的直方图并不能实现直方图完全均衡化，而仅能在许可范围内尽可能接近于均匀概率密度函数，这是由于灰度级的不连续和像素的不连续所造成的。

图 5-11(c)表明，尽管没有达到理论上的完全均衡化效果，但目标图像直方图比原始图像直方图已有很大的改善。原始图像灰度值偏低，图像整体上偏暗，直方图均衡化后，其亮度得到了较大的提升，灰度值分布比较均衡。

通过本例还可以得出，灰度级由 8 个缩减成为 5 个，经过直方图均衡化运算以后，变换后的灰度级数量减少了，这是由于均衡化过程中要进行舍入运算造成的，这种情况称为“简并”现象。通常情况下，被简并的是原始图像中频数较低(像素点较少)的一些灰度级。若这些灰度级构成的图像细节比较重要，则可以采用局部自适应直方图均衡化技术，也可以采用增加像素位数(如以 12 位灰度图像代替 8 位灰度图像)的方法来减少由于灰度级简并现象所造成的灰度级数量损失。

5.4.3 直方图规定化

理想情况下，直方图均衡化实现了图像灰度的均衡分布，对提高图像对比度、提升图像亮度具有明显的作用。在实际应用中，有时并不需要图像的直方图具有整体的均匀分布，而希望直方图与规定要求的直方图一致，这就是直方图规定化。它可以人为地改变原始图像直方图的形状，使之成为某个特定的形状，即增强特定灰度级分布范围内的图像。

1. 基本原理

直方图规定化的目的就是调整原始图像的直方图使之符合某一规定直方图的要求。设 $P_r(r)$和 $P_z(z)$分别表示原始灰度图像和目标图像的灰度分布概率密度函数，根据直方图规定化的特点与要求，应使原始图像的直方图具有 $P_z(z)$所表示的形状。因此，建立 $P_r(r)$和 $P_z(z)$之间的关系是直方图规定化必须解决的问题。

根据直方图均衡化理论，首先对原始图像进行直方图均衡化处理，即求变换函数：

$$s = T(r) = \int_0^r P_r(x)\mathrm{d}x \tag{5-15}$$

现假定直方图规定化的目标图像已经实现，因此，对于目标图像也采用同样的方法进行均衡化处理，因而有

$$v = G(z) = \int_0^z P_z(x)\mathrm{d}x \tag{5-16}$$

上式的逆变换为

$$z = G^{-1}(v) \tag{5-17}$$

上式表明，可通过均衡化后的灰度级 v 求出目标图像的灰度级 z。由于对目标图像和原始图像都进行了均衡化处理，因此具有相同的分布密度，即

$$P_s(s) = P_v(v)$$

因而可以用原始图像均衡化以后的灰度级 s 代替 v，即

$$z = G^{-1}(v) = G^{-1}(s) \tag{5-18}$$

所以，可以依据原始图像均衡化后的图像的灰度值得到目标图像的灰度级 z。

2. 直方图规定化方法

根据上述理论推导，可以得出直方图规定化处理的一般步骤如下：

(1) 根据直方图均衡化原理，对原始图像进行直方图均衡化处理。

(2) 按照目标图像的概率密度函数 $P_z(z)$，求解目标图像进行均衡化处理的变换函数 $G(z)$。

(3) 利用原始图像均衡化中得到的灰度级 s 代替 v，求解逆变换 $z=G^{-1}(s)$。

经过上述处理得到的目标图像的灰度级将具有事先规定的概率密度 $P_z(z)$。上述变换过程中所包含的两个变换函数 $T(r)$ 和 $G^{-1}(s)$ 可形成复合函数，即可表示为

$$z = G^{-1}(s) = G^{-1}[T(r)] \tag{5-19}$$

由此可知，无须进行直方图均衡化运算就可以直接实现直方图规定化处理，通过复合函数关系有效简化了直方图规定化处理过程，求出 $T(r)$ 和 $G^{-1}(s)$ 之间的复合函数关系就可以直接对原始图像进行变换。

3. 直方图规定化的应用

例 5-4：直方图规定化计算原理与过程

为便于与直方图均衡化进行对比，仍采用前述直方图均衡化相同大小的 64×64 灰度图像数据，如表 5-1 所示，共有 8 个灰度级，根据直方图规定化技术，按表 5-2 给定数据对原始图像进行直方图规定化处理。

图 5-12(a) 为规定化直方图，图中横坐标轴标记 0、1、…、7 分别代表 0、1/7、…、1。

表 5-2　64×64 大小图像灰度分布表

z_k	$P_z(z_k)$
$z_0=0$	0.00
$z_1=1/7$	0.00
$z_2=2/7$	0.00
$z_3=3/7$	0.15
$z_4=4/7$	0.20
$z_5=5/7$	0.30
$z_6=6/7$	0.20
$z_7=1$	0.15

根据直方图规定化的原理，其处理过程如下：

（1）原始图像直方图均衡化。

直方图规定化处理需首先对原始图像进行直方图均衡化处理，由于与例 5-3 数据相同，因此可以直接利用直方图均衡化的处理结果。

（2）求变换函数。

根据离散数字图像概率密度变换函数的公式计算变换函数，即

$$v_k = G(z_k) = \sum_{j=0}^{k} P_z(z_j)$$

由此可得

$$v_0 = G(z_0) = \sum_{j=0}^{0} P_z(z_j) = P_z(z_0) = 0.00$$

$$v_1 = G(z_1) = \sum_{j=0}^{1} P_z(z_j) = P_z(z_0) + P_z(z_1) = 0.00$$

$$v_2 = G(z_2) = \sum_{j=0}^{2} P_z(z_j) = P_z(z_0) + P_z(z_1) + P_z(z_2) = 0.00$$

$$v_3 = G(z_3) = \sum_{j=0}^{3} P_z(z_j) = P_z(z_0) + P_z(z_1) + P_z(z_2) + P_z(z_3) = 0.15$$

以相同的方法可分别求出

$$v_4 = G(z_4) = \sum_{j=0}^{4} P_z(z_j) = 0.35$$

$$v_5 = G(z_5) = \sum_{j=0}^{5} P_z(z_j) = 0.65$$

$$v_6 = G(z_6) = \sum_{j=0}^{6} P_z(z_j) = 0.85$$

$$v_7 = G(z_7) = \sum_{j=0}^{7} P_z(z_j) = 1.00$$

以上数据表明了 v_k 与 z_k 之间的变换关系，根据这些数据可得出 v_k 与 z_k 之间的离散函数关系，如图 5-12(b)所示。实际上，这些数据同样反映了 v_k 与 z_k 之间的逆变换关系，为以 s_k 代替 v_k 奠定了基础。

（3）求 $z_k = G^{-1}(s_k)$。

以 s_k 代替 v_k，进行 $z_k = G^{-1}(v_k) = G^{-1}(s_k)$ 的逆变换运算，求出最终的灰度级 z_k。在离散情况下，逆变换需进行近似处理。例如，最接近于 $s_0 = 1/7 = 0.14$ 的是 $v_3 = 0.15$，因此可用 s_0 代替 v_3，进行反变换 $G^{-1}(s_0) = z_3$，即 s_0 映射到灰度级 z_3。因此，可得到表 5-3 所示映射关系。

表 5-3　s_k 与 z_k 映射关系表

s_k	z_k
$s_0 = 1/7$	$z_3 = 3/7$
$s_1 = 3/7$	$z_4 = 4/7$
$s_2 = 5/7$	$z_5 = 5/7$
$s_3 = 6/7$	$z_6 = 6/7$
$s_4 = 1$	$z_7 = 1$

(4) 求 r_k 和 s_k 的映射关系。

根据步骤(1)中得到的 r_k 和 s_k 之间的数值对应关系可得到 r_k 和 z_k 之间的映射关系，如表 5-4 所示。

表 5-4　r_k 与 z_k 映射关系表

r_k	z_k
$r_0=0$	$z_3=3/7$
$r_1=1/7$	$z_4=4/7$
$r_2=2/7$	$z_5=5/7$
$r_3=3/7$	$z_6=6/7$
$r_4=4/7$	$z_6=6/7$
$r_5=5/7$	$z_7=1$
$r_6=6/7$	$z_7=1$
$r_7=1$	$z_7=1$

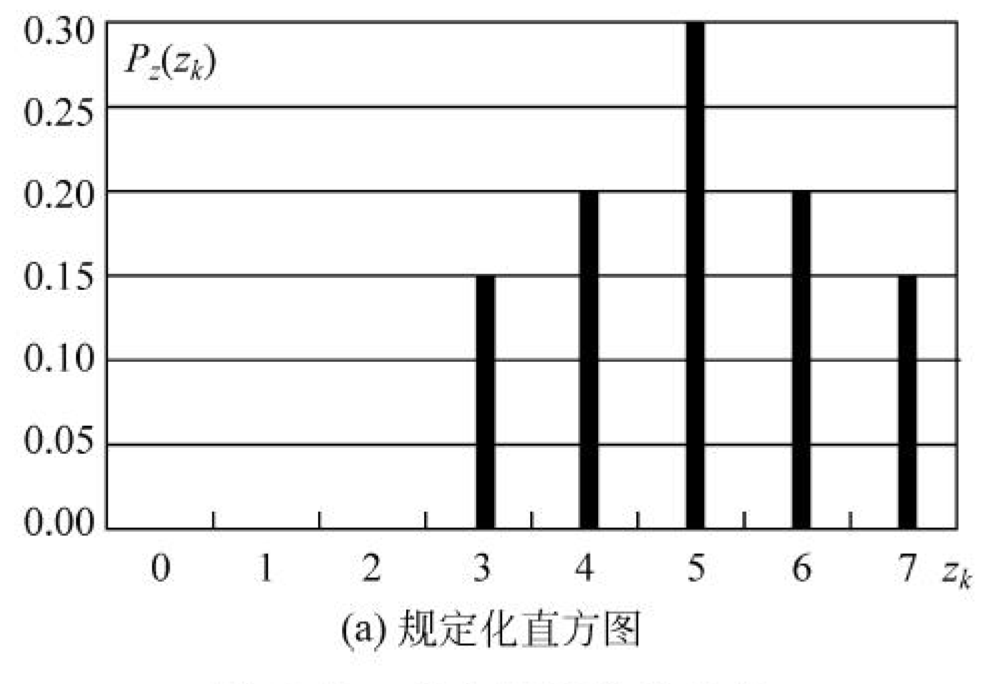
(5) 绘出规定化处理直方图。

根据表 5-3 和表 5-4 所示的映射关系重新计算各灰度级的像素点的数量，计算结果如表 5-5 所示，完成对原始图像直方图规定化运算的最终处理。图 5-12(c)为对原始图像进行直方图规定化运算后的结果直方图。

根据本例的实际运算结果可知，与直方图均衡化的情况类似，直方图规定化的运算结果与期望直方图(即规定直方图)之间也存在一定的差异，这同样是由于从连续到离散的转换引入了离散误差的原因，而且这种误差与原始图像灰度级数量有关，通常情况下，灰度级越少，误差越大。但尽管存在误差，直方图规定化对图像亮度和对比度的提高依然具有很明显的效果。

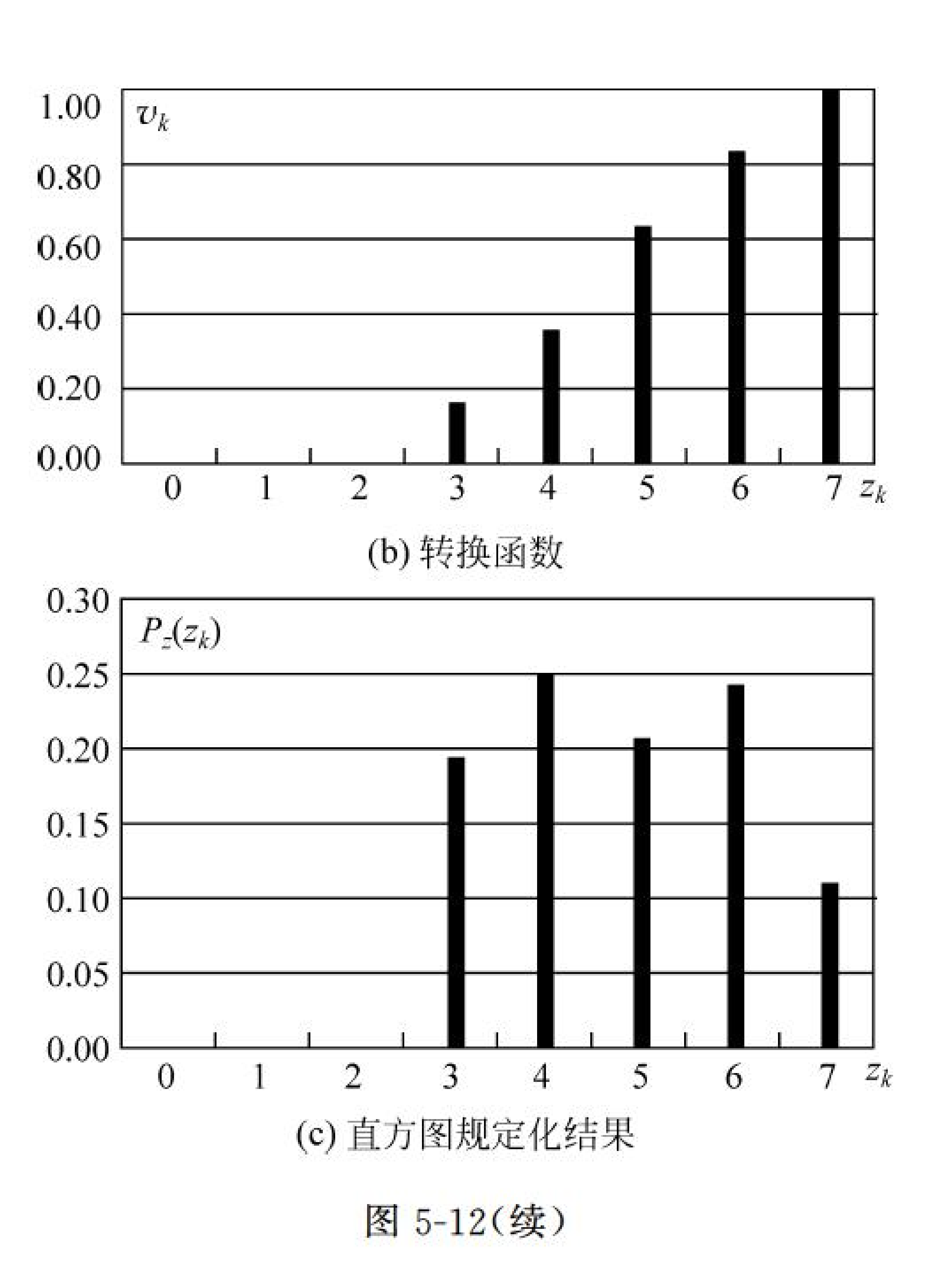
表 5-5　结果直方图统计数据

z_k	n_k	$P_z(z_k)$
$z_0=0$	0	0.00
$z_1=1/7$	0	0.00
$z_2=2/7$	0	0.00
$z_3=3/7$	790	0.19
$z_4=4/7$	1023	0.25
$z_5=5/7$	850	0.21
$z_6=6/7$	985	0.24
$z_7=1$	448	0.11



(a) 规定化直方图

图 5-12　直方图规定化过程



(b) 转换函数

(c) 直方图规定化结果

图 5-12(续)

5.5　代数运算增强

第 4 章已介绍了图像的代数运算，实际上，代数运算也可以对数字图像进行增强。基于代数运算的图像增强方法是三类灰度变换增强法中的最后一类方法。

代数运算图像增强是指通过对多幅图像进行某一种或几种代数运算增强图像的某些特定信息，从而达到改善图像质量和效果的目的。代数运算是指对两幅或两幅以上图像的相应像素点进行加、减、乘、除及其复合运算而生成一幅新的图像。新图像与原始图像相比，能有效地突出某些专题信息。

5.5.1　代数运算增强的应用

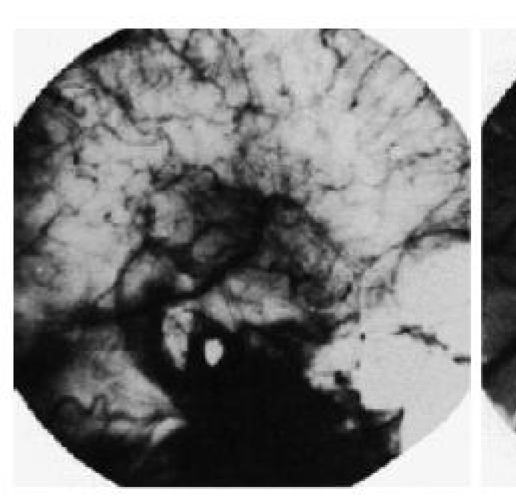
具体而言，代数运算用于图像增强包括以下几方面。

(1) 通过对同一景物或物体的多幅图像进行求平均值运算，理论上可以起到降低或消除图像加性随机噪声的作用。

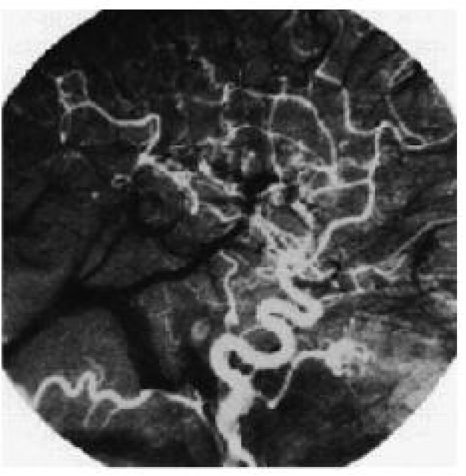
(2) 图像间的差值运算可以获取图像的时间信息。两幅不同时期的遥感图像间的减法运算可以得到相应时间段内的动态信息，即从前次拍摄到第二次拍摄期间该场景或其中某些物体的变化情况。遥感图像的相减运算主要用于提取多时相遥感图像中的因时间迁移所发生的变化信息(即动态信息)。例如，通过图像的减法运算可以获得土壤荒漠化的动态变化情况、河流湖泊在汛期和干旱季节的水位变化情况；对同一地区或场景冬、夏两季的图像进行差值运算可以得到植被信息变化情况，并可以得到常绿树种的覆盖率数据以及用于对某些特定树种进行辨识；通过对一个城市不同年间的卫星遥感图像进行差值运算，可以了

解城市的发展规模、扩展速度及耕地占用情况等信息，从而获得城市土地控制方面的宏观信息。

此外，在医学应用等特殊领域，由于医学检查的需要，医生只关心指定的检查部位、组织或器官的病理信息，而拍摄过程中却不可能将肌肉脂肪等相关组织进行分离，这时这些组织就成为了图像的背景，这些背景信息与诊断目的是不相关的，甚至对被诊断的对象具有不利影响，因此有必要消除或削弱这些背景信息的作用。如图 5-13 所示，通过消除检查部位的背景，可以获得较理想的诊断部位图像信息。



(a) 含有背景的原始图像



(b) 消除背景的增强图像

图 5-13 代数运算图像增强

需要注意的是，尽管图像的差值运算算法简单，但有时减法运算可能会产生负像素值，因此应特别注意并对产生的负值进行适当处理。而差值图像的显示，一般都要经过适当的灰度调整，使差值为 0 的像素点的灰度值映射到灰度值范围的中值，例如对于 8 位像素图像，中值为 128，有时也可以将 8 位像素图像的差值图像保存为 16 位的图像，获得更加细微的动态变化信息。

（3）图像的卷积运算（乘法运算）可以对图像的边缘轮廓进行增强。

（4）图像间的除法运算（通常为比值运算）可以消除地形对遥感图像的影响。

5.5.2 比值处理与应用

图像的比值处理是指将两个波段的灰度图像对应像素点的灰度值进行除法运算，可推广定义为各波段数据及其代数组合后进行比值运算。比值所反映的不是地物光谱反射率曲线的特征，而是波段间光谱反射曲线斜率变化的特征，它可使那些反射率相似而光谱反射曲线斜率不同的地物类别得以区分。因为比值和照度无关，所以比值又可以消除地形及其他因素造成的光照条件差异的影响。比值运算属于图像代数运算的一种，常用于多光谱卫星遥感图像处理等方面。比值处理的图像不是单个的灰度图像，而是多波段的灰度图像，即必须是对多个图像进行处理。由于通过卫星遥感可以获得同一区域的多个不同波段的图像，采用比值处理可以扩大不同地物的灰度值差异，消除或减弱地形朝向和云层等因素对图像的干扰，因此比值处理在遥感遥测中具有重要作用。

1. 比值处理原理

根据物理光学原理，图像中各像素点的灰度值可以理解为景物的入射光强（照明分量）和反射光强（反射分量）之乘积。同一地区的多光谱卫星图像，由于各个波段几乎是同时成像，所以各个波段图像的入射光强近似相等，波段间的比值可以消除入射光强对灰度值的作用。反射光强可以在一定程度上反应地物差异，经比值处理后，物体反射光强的差异将扩

大,从而有利于地物的识别。有些地物对同一波段的光具有相同或相似的反射率,两者间的差异较小,但其对不同波段光的反射率差异却很大。如表5-6所示,在TM4波段,反射率分别为16和17;在TM7波段,反射率分别为1和4。因此,对于水和沙滩,无论在TM4还是TM7波段,图像的灰度值差异都较小,但两类地物对TM4和TM7这两个不同波段光的反射率差别却很大,因此,经过对水和沙滩的TM4与TM7不同波段图像的比值处理可以充分反映这一差距。

表5-6　波段反射率差异

波　　段	水　　体	沙　　滩
TM4	16	17
TM7	1	4
TM4/TM7	16	4.25

2. 比值处理在遥感图像的应用

比值处理在遥感图像中的应用非常广泛,其中典型的应用包括用于解决"同谱异物"问题、"同物异谱"问题以及用于植被指数的测量等。

1)"同谱异物"问题

所谓"同谱异物"问题,是指尽管图像具有相同的频谱特性,但却是不同的物体。应用比值处理扩大不同地物的光谱差异,有些地物在单波段图像中的差异较小,通过选择适当的波段进行比值处理,可以有效扩大不同地物之间的差异,解决图像中存在的"同谱异物"问题。比值波段的合理选择应充分考虑不同波段特性和不同的地面物体群等综合因素的影响,不能一概而论,例如对区分某些地物有效的两个比值波段不一定适合选择为另一类地物的比值波段。

2)"同物异谱"问题

所谓"同物异谱"问题,是指相同的物体,但却表现为不同的频谱特性。光线照射所形成的阴影面对卫星图像的地物识别和信息提取具有较大影响,阴影的形成既包括诸如山坡朝向等地形因素,也包括浓密云雾层等因素的影响。由于阴影的影响,可能导致阴坡与阳坡的同一类地面物体被误认为是两类物体,即产生"同物异谱"现象。比值处理可以消除地形的影响,使处在不同坡向的同一类地面物体的灰度值趋向一致。

3)植被指数测量

20世纪80年代至90年代,我国曾应用植被指数(vegetation index)进行了广泛的地表覆盖分析研究,植被指数测量是比值处理在遥感图像中最成功的应用之一。植被指数作为一种简单而有效的参考量,在资源环境的遥感遥测中有着广泛的应用。由于光合作用的影响,植被在红光波段形成一个吸收带,在近红外波段形成一个反射带。因此,理论上可以通过红外与近红外波段的比值信息反映植被信息,即可以进行植被指数的测量。由于植被指数与大气效应、土壤成分、植被本身等因素密切相关,所以它具有一定的地域特征和时效特征。最常用的形式则是归一化植被指数(NDVI),它表示为近红外波段与红光波段的差与两者的和的比值。以陆地卫星TM为例,可以表示为

$$\mathrm{NDVI} = C\frac{\mathrm{TM4} - \mathrm{TM3}}{\mathrm{TM4} + \mathrm{TM3}} \tag{5-20}$$

式中，C 是需要根据具体情况确定的常数；TM3 和 TM4 分别表示红光波段与近红外波段。

此外，植被指数还有多种其他形式，如加权差值植被指数、垂直植被指数、土壤修正植被指数等。

尽管比值处理在扩大地物间的灰度差异、消除阴影影响等方面具有积极作用，但在对遥感图像进行比值处理时应注意以下问题：

(1) 植被指数计算公式虽然简单，但其波段的选择及相关参数的调整需要经验并可能需要反复多次实验（如归一化植被指数中常数 C 的确定），以便对地物的光谱特征进行测定与分析，并根据分析的结果调整相关的参数。

(2) 与原始图像相比，比值处理生成的图像不具有独立的波谱意义，而且还会丧失图像的一些地形信息，因此在对比值图像进行彩色合成时，一般要增加 TM5 波段，以便能反映地形信息。

(3) 比值处理在增强地物间差异的同时，某些情况下，会使图像中的噪声信息放大，因此，必要时应先进行低通滤波，对图像进行平滑处理后，再进行比值处理。

5.6 空域滤波增强

一幅图像通常记录了物体的空间和光谱等多种信息，对于常见的灰度图像，其光谱信息是以图像各像素点的灰度值来记录的，各种灰度变换方法都可以达到增强光谱信息的目的。图像间的差值运算可以获取图像的时间信息，即动态信息。图像的空间纹理信息增强则可以通过空域滤波和频域滤波方法实现。

图像的空间纹理信息可以反映图像中物体的位置、形状和大小等特征。例如，一幅图像内物体的边缘轮廓部分一般灰度值变化较大，因此呈现出高频特征，而一个比较平滑的物体内部，由于灰度值相对均匀则呈现低频特征。

数字图像处理根据需要可以分别采取不同方法对图像的高频和低频特征信息进行增强。例如，公安系统在进行人像的比对查询时，就需要突出高频信息增强五官轮廓特征信息，这样不仅可以提高比对速度，还可以提升比对的准确率。增强图像的高频信息称为高通滤波，它可以突出物体的边缘轮廓，从而起到锐化图像的作用，因此又称为锐化滤波器。根据频域分析理论，高通滤波能起到减弱或消除图像的低频分量、保留高频分量的作用。相应地，低通滤波则是指对图像的低频信息进行增强，又称为平滑滤波器。它可以实现图像的平滑处理，一般用于数字图像的噪声消除。平滑滤波可以减弱甚至消除图像的高频分量、保留低频分量。

5.6.1 空域滤波模板

数字图像空域滤波的实现是应用模板卷积方法对每一像素点的邻域进行处理完成的，它可分为线性和非线性两类空域滤波方法。模板（template）运算方法是数字图像处理中常用的一种运算方式，图像的平滑、锐化、细化和边缘检测等都运用到模板操作。例如，常见的平滑算法是将原始图中的一个像素点的灰度值和它周围邻近其他 8 个像素的灰度值相加，以平均值作为目标图像中该像素的灰度值。该运算可用模板方法表示为

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

采用以上形式模板就可以实现上述算术平均运算的功能。模板操作实现了一种邻域运算,即某个像素点的结果不仅和本像素灰度值有关,而且和其邻域点的值有关。模板运算的数学含义是卷积(或互相关)运算。

卷积是一种用途很广的算法,可用卷积来完成各种处理变换,图 5-14 说明了 3×3 邻域与 3×3 模板(又称为卷积核)卷积运算的处理过程。邻域中的每一个像素点 $f_1, f_2, \cdots, f_9$ 分别与模板中的每一个元素 $h_1, h_2, \cdots, h_9$ 相乘,乘积求和所得的结果即为中心像素点的灰度值。

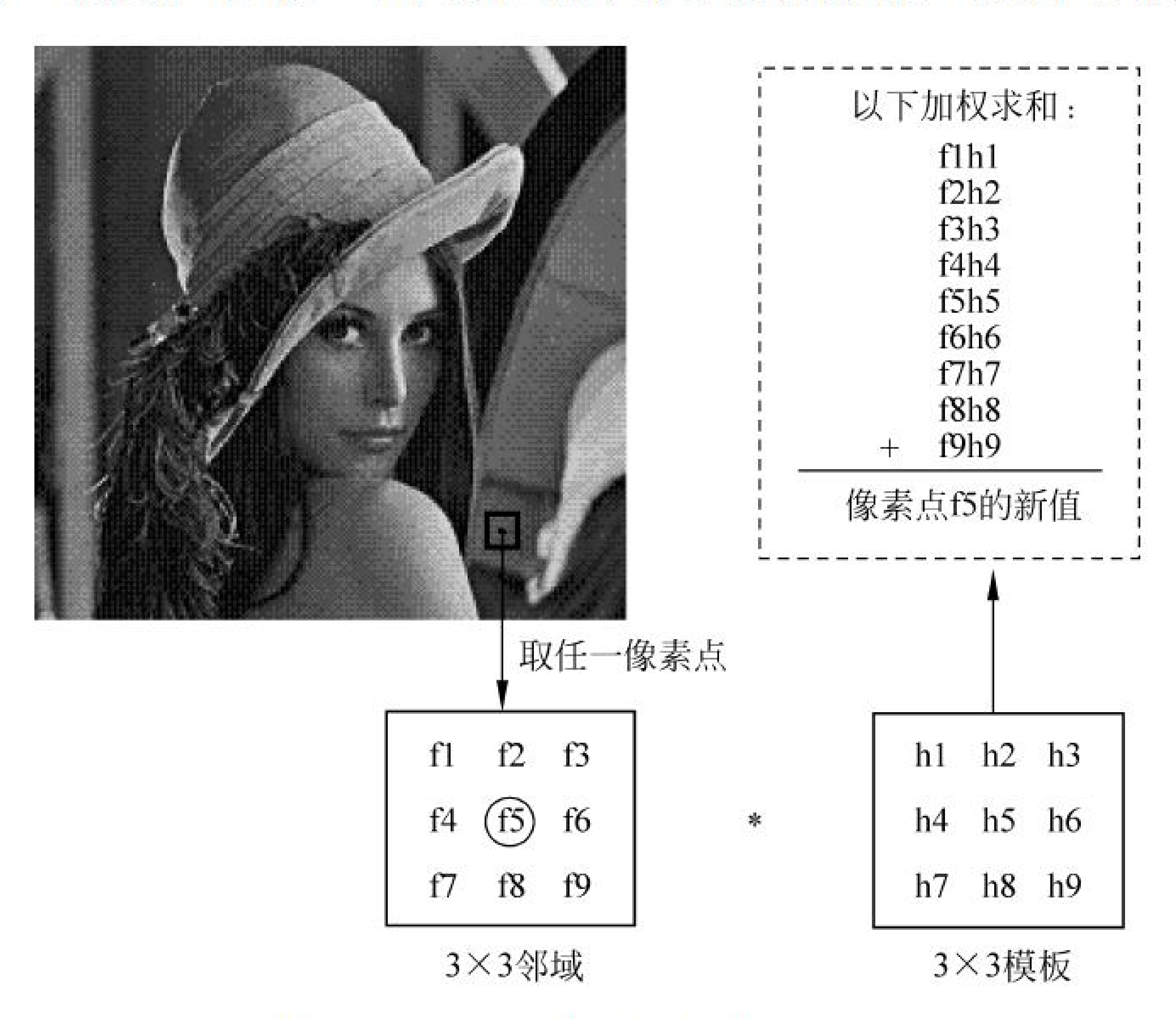


图 5-14　空域滤波模板卷积运算过程

无论采用什么样的滤波器,也不管是高通或低通滤波器,其计算步骤基本类似。具体过程一般如下:

(1) 将模板在图像中依次移动,并将模板中心与每个像素点依次重合(不包括边缘像素点)。

(2) 将模板中的各系数与其对应的像素点相乘,并将所有结果相加(或进行其他运算)。

(3) 加权值即为目标图像中对应模板中心位置的像素点灰度值。

模板或卷积的加权运算中,以下问题需要引起注意:

(1) 图像边界问题。当在图像上移动模板至图像的边界时,在原图像中找不到与卷积核中的加权系数相对应的 9 个像素,即卷积核悬挂在图像缓冲区的边界上,这种现象在图像的上、下、左、右四个边界上均会出现。一般解决这个问题有两种简单方法:一种方法是忽略图像边界数据,即不处理图像的边缘像素;另一种方法是在图像四周复制原图像边界像素的值(或其他方法),从而达到平滑边界像素点的目的。实际应用中,多采用第一种方法。开发软件时,循环次数应小于图像的行列数,对于大小为 $N\times N$ 的图像,一般可取 $N-2$,对于图像边界处的像素点可以通过边界拓展的方法进行处理。

(2) 卷积求和计算出来的像素值的动态范围问题,对此可简单地将其值置为0或255。

(3) 模板的大小与平滑效果直接相关,模板越大,平滑的效果越明显,但模板太大会造成图像的边缘信息的损失,模板的大小要在保证消除噪声的前提下尽可能保持图像的边缘信息。开发图像应用软件时,一般通过设置对话框的方式,使用户可以灵活选择甚至自定义模板的大小及模板系数。模板大小一般为奇数,且不超过11×11。

5.6.2 线性平滑滤波

各种线性平滑滤波方法一般都基于模板卷积进行运算,平滑滤波一般用于消除图像中的随机噪声,起到图像平滑的作用。线性平滑滤波的作用是对图像的高频分量进行削弱或消除,增强图像的低频分量。平滑滤波器的设计比较简单,若模板各系数取不同的值,就可得到不同的平滑滤波器。

邻域均值滤波法和中值滤波法是常用的平滑滤波方法,其中,邻域平均法是线性运算,而中值滤波法是非线性运算。

1. 邻域平均法

邻域平均滤波法是将一个像素点及其邻域中的所有像素点的平均值赋给输出图像中相应的像素点,从而达到平滑的目的,又称均值滤波法。最简单的邻域平均法是所有模板系数都取相同的值,例如,取模板系数为1,这种模板又称为Box模板,常用的模板为3×3和5×5的两种模板类型。

3×3模板:

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

5×5模板:

$$\frac{1}{25}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1\end{bmatrix}$$

邻域平均滤波法的运算公式为

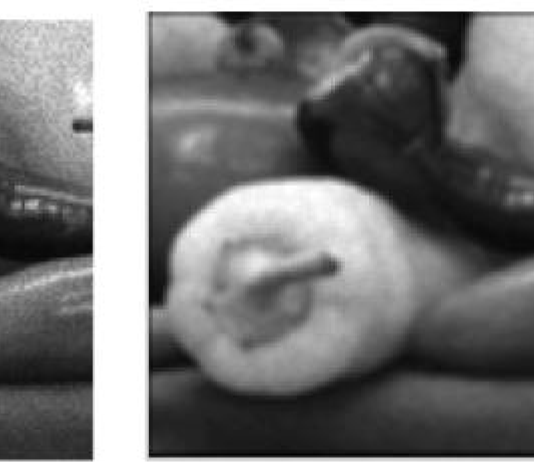
$$g(x,y)=\frac{1}{N}\sum_{j\in M}f(i,j)\quad x,y=0,1,2,\cdots,N-1 \tag{5-21}$$

式中,M是以$(x,\ y)$为中心的邻域像素点的集合;N是该邻域内像素点总数。

对每个像素点按该公式进行计算即可得到增强图像中所有像素点的灰度值。Box模板对当前像素及其相邻的像素点都一视同仁,统一进行平均处理,可以滤去图像中的随机噪声等。例5-5的图像实验也说明了这一点。

例5-5:邻域平均滤波增强图像示例

图5-15为邻域平均滤波增强图像实验。图5-15(a)为含有随机噪声的灰度图像,图5-15(b)、图5-15(c)、图5-15(d)是分别用3×3、5×5、7×7模板进行平滑滤波所得到的图像。通过该图也可以看出,邻域平均法对平滑图像具有一定的效果。


(a) 噪声图像

(b) 3×3模板

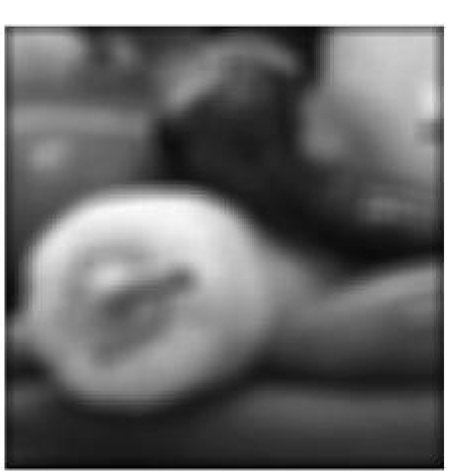
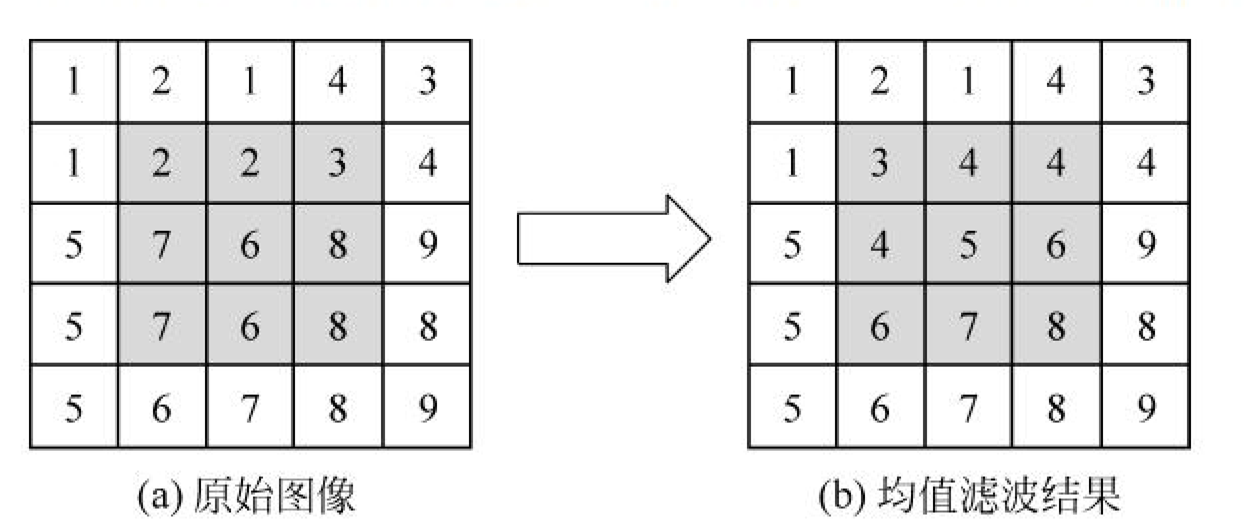
(c) 5×5模板

(d) 7×7模板

图 5-15　邻域平均法的增强效果

例 5-6：邻域平均滤波计算过程示例

如图 5-16(a)所示为原始图像数据，对该图像采用 3×3 模板进行邻域平均滤波。



1	2	1	4	3
1	2	2	3	4
5	7	6	8	9
5	7	6	8	8
5	6	7	8	9

(a) 原始图像

1	2	1	4	3
1	3	4	4	4
5	4	5	6	9
5	6	7	8	8
5	6	7	8	9

(b) 均值滤波结果

图 5-16　3×3 模板平滑处理过程

根据邻域平均法滤波的原理，采用 3×3 模板对图 5-16(a)进行滤波(边缘像素点除外)，以第 2 行第 2 列的像素为例，滤波以后该像素点的灰度值为

$$\frac{1}{9}\times(1\times1+2\times1+1\times1+1\times1+2\times1+2\times1+5\times1+7\times1+6\times1)=3$$

对于第 2 行第 3 列的像素，滤波以后该像素点的灰度值为

$$\frac{1}{9}\times(2\times1+1\times1+4\times1+2\times1+2\times1+3\times1+7\times1+6\times1+8\times1)\approx4$$

同样的方法，可以计算其他各点的灰度值。对于各点的灰度值需进行四舍五入处理。滤波结果如图 5-16(b)所示。

2. 加权平均法

邻域平均法认为被处理像素与邻域内其他像素的作用相等，因此，模板的系数完全相等。加权均值滤波法认为，对于被处理的像素点，其邻域内像素的贡献是不一样的，离被处理像素距离近者，其贡献大，因此，模板的系数大小不完全相等。以 3×3 模板为例，常用的加权平均模板矩阵为

$$\frac{1}{21}\times\begin{bmatrix}1 & 4 & 1\\ 4 & 1 & 4\\ 1 & 4 & 1\end{bmatrix} \tag{5-22}$$

3. 高斯滤波

高斯滤波(Gauss filter)实质上是根据高斯函数计算出来的一种滤波模板，其用途为对图像信号进行平滑消噪处理。图像处理的后期应用，其噪声是一个很重要的问题，因为误差会累计传递，高斯平滑滤波器对于抑制服从正态分布的噪声非常有效，用于得到信噪比 SNR 较高

的图像(反映真实信号)。高斯滤波对图像邻域内像素进行平滑时,邻域内不同位置的像素被赋予不同的权值,对图像进行平滑的同时,能够更多地保留图像的总体灰度分布特征。

数值图像处理中,高斯滤波主要可以使用两种方法实现。一种方法是离散化窗口滑窗卷积,另一种方法是通过傅里叶变换。一般采用第一种离散化窗口卷积形式实现,5×5的高斯模板为

$$\frac{1}{273}\times\begin{bmatrix}1 & 4 & 7 & 4 & 1\\ 4 & 16 & 26 & 16 & 4\\ 7 & 26 & 41 & 26 & 7\\ 4 & 16 & 26 & 16 & 4\\ 1 & 4 & 7 & 4 & 1\end{bmatrix} \tag{5-23}$$

5.6.3 非线性平滑滤波

虽然基于模板的线性滤波方法简便易行,但对于既需要消除噪声又要保持好的图像细节时,其效果并理想,而非线性滤波在这方面具有一定的优势。非线性滤波最经典适用的方法是基于模板的统计排序滤波,主要包括中值滤波、最大值滤波和最小值滤波三种方法,该滤波方法基于滤波器所在图像区域中像素的排序,由排序结果决定的值代替中心像素的值。

1. 中值滤波

尽管邻域平均滤波法可以起到平滑图像的作用,但在消除噪声的同时,会使图像中的一些细节变得模糊。中值滤波法则在消除噪声的同时还能保持图像中的细节部分,防止图像的边缘部分模糊。中值滤波法于1971年开始逐步由一维时间序列推广运用到二维图像信号处理,并发挥了较好的作用。与邻域平均法不同,中值滤波是一种非线性滤波,它将邻域内所有像素点值从小到大排序,取中间值作为中心像素点的输出值。采用中值滤波,在一定的条件下,可以克服数字图像细节模糊的问题。

中值滤波的原理是利用一个奇数点的移动窗口,将窗口中心点的值用窗口各点的中间值代替。

设($x_{ij}(i,j)\in I^2$)表示图像中各像素的灰度值,滤波窗口为W的二维中值滤波可定义为

$$g_{ij} = \underset{W}{\mathrm{Med}}\{x_{ij}\} \tag{5-24}$$

例如,若窗口内有5点,其值分别为118、88、98、128、158,则窗口内各点的中值即为118。中值滤波与均值滤波不同,它不是通过对邻域内的所有像素点求平均值来消除噪声的,而是让与周围像素点灰度值的差比较大的像素点改取近似于周围像素点灰度值的值,从而达到消除噪声的目的。

例5-7:中值滤波增强图像示例

本实验说明了不同大小模板的中值滤波的平滑效果,图5-17(a)为含有随机噪声的灰度图像,5-17(b)、5-17(c)、5-17(d)是分别用3×3、5×5、7×7模板得到的平滑图像。

从图5-17与图5-15的比较可以看出,中值滤波一定程度上保持了图像的边缘轮廓,图像比较清晰,其效果比均值滤波要好,其中以3×3模板效果最佳,7×7模板虽然消除了噪声,但图像内部的边缘信息也变得模糊了。


(a) 噪声图像


(b) 3×3模板


(c) 5×5模板


(d) 7×7模板

图 5-17　中值滤波的效果

例 5-8：中值滤波计算示例

如图 5-18(a)所示为原始图像数据，采用 3×3 模板对该图像进行中值滤波。

1	2	1	4	3
1	2	2	3	4
5	7	6	8	9
5	7	6	8	8
5	6	7	8	9

(a) 原始图像

3×3窗中值滤波

边界像素不处理

1	2	1	4	3
1	2	3	4	4
5	5	6	6	9
5	6	7	8	8
5	6	7	8	9

(b) 中值滤波结果

图 5-18　中值滤波处理方法

根据中值滤波的原理，采用 3×3 模板对图 5-18(a)进行滤波(边缘像素点除外)，可以对图像进行平滑处理。以第 3 行第 3 列的像素为例，滤波以后该像素点的灰度值为

$$\mathrm{Med}\begin{pmatrix}2 & 2 & 3\\ 7 & 6 & 8\\ 7 & 6 & 8\end{pmatrix}=6$$

以同样的方法，可以得出除边缘点外的其他像素点的滤波值。中值滤波结果如图 5-18(b)所示。

2. 最大值滤波

最大值滤波法的实现过程与中值滤波类似，不同之处是用像素邻域内的最大值代替被处理像素的灰度值实现滤波。其主要用途是寻找最亮点。

计算公式为

$$g_{ij}=\underset{W}{\mathrm{Max}}\{x_{ij}\} \tag{5-25}$$

3. 最小值滤波

最小值滤波法是用像素邻域内的最小值代替被处理像素的灰度值，从而实现滤波。最小值滤波器主要用于寻找最暗点。

计算公式为

$$g_{ij}=\underset{W}{\mathrm{Min}}\{x_{ij}\} \tag{5-26}$$

5.6.4　锐化滤波器

图像的平滑处理会使图像的边缘纹理信息受到损失，图像变得比较模糊。如果需要突出图像的边缘纹理信息，则可以通过锐化滤波器实现，它可以消除或减弱图像的低频分量从

而增强图像中物体的边缘轮廓信息，使得除边缘以外的像素点的灰度值趋向于零。常用的锐化滤波主要有梯度法、拉普拉斯算子法和定向滤波等。

锐化滤器是以对图像的微分运算或差分运算为基础的。微分运算是求像素点灰度值的变化率，而图像内不同物体边缘处的像素点的灰度值往往变化比较明显，因此微分运算或差分运算可以起到增强边缘信息的作用。

1. 梯度法

1）连续梯度算子

对于二维连续系统，在坐标位置(x,y)处的梯度向量为

$$G[f(x,y)] = \mathrm{i}\frac{\partial f}{\partial x} + \mathrm{j}\frac{\partial f}{\partial y} \tag{5-27}$$

$G[f(x,y)]$是一个矢量，它指向$f(x,y)$的最大变化率方向，一阶偏导数$\frac{\partial f}{\partial x}$和$\frac{\partial f}{\partial y}$分别表示$f(x,y)$沿$x$方向和$y$方向的灰度变化率。由于梯度是向量，因此其幅值为

$$|G[f(x,y)]| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \tag{5-28}$$

梯度的方向为

$$\theta = \arctan\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right) \tag{5-29}$$

2）离散梯度算子

由于无论x方向还是y方向，离散系统的坐标值最小增量为1，因而以相邻点之差近似表示梯度分量。

$$\begin{cases} \Delta_x f = f(m,n) - f(m-1,n) \\ \Delta_y f = f(m,n) - f(m,n-1) \end{cases} \tag{5-30}$$

梯度的幅值为

$$\nabla f = \sqrt{(\Delta_x f)^2 + (\Delta_y f)^2} \tag{5-31}$$

在某些情况下，为简单起见(避免平方根运算)，可以根据具体情况分别采用如下方法计算梯度的近似值：

$$\nabla f \approx |\Delta_x f| + |\Delta_y f| \tag{5-32}$$

或

$$\nabla f \approx \max(|\Delta_x f|, |\Delta_y f|) \tag{5-33}$$

显然，离散系统梯度幅值与上述两种近似值之间存在以下关系：

$$\max(|\Delta_x f|, |\Delta_y f|) \leqslant \sqrt{(\Delta_x f)^2 + (\Delta_y f)^2} \leqslant |\Delta_x f| + |\Delta_y f| \tag{5-34}$$

运用差分算子，罗伯特(Robert)梯度算子是常用的梯度差分法，可以表示为

$$G[f(x,y)] = |f(i,j) - f(i+1,j+1)| + |f(i+1,j) - f(i,j+1)| \tag{5-35}$$

罗伯特梯度算子可以用如图5-19所示的几何图形表示，该算子实际上是一种交叉差分运算法则。显然，采用罗伯特等梯度算子时，在图像的最后一行和最后一列无法计算，这时可以采用前一行或前一列的梯度值近似代替。罗伯特梯度算子、索贝尔算子等在图像

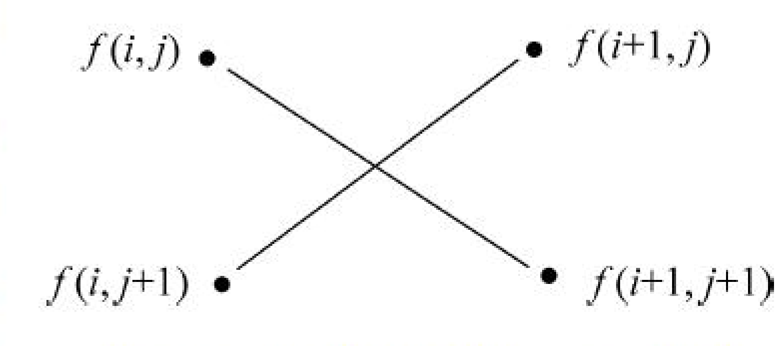


图5-19 罗伯特梯度算子运算图

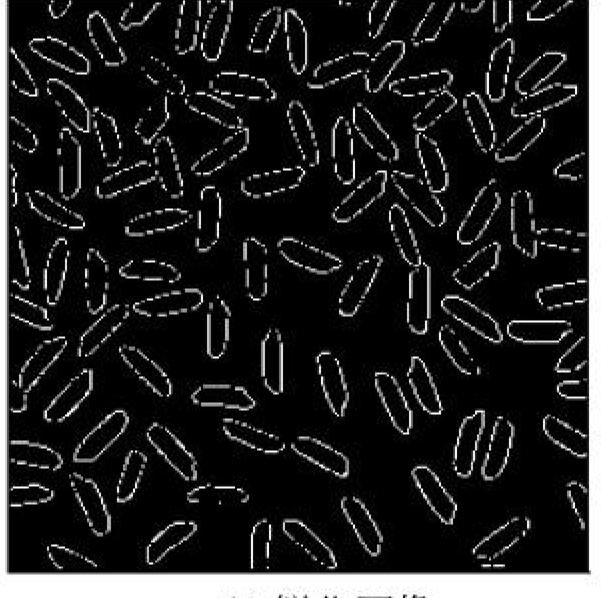
的边缘检测中具有广泛运用。

例 5-9：锐化滤波的边缘增强示例

本实验为图像锐化增强边缘信息效果实验。图 5-20(a)为米粒的原始图像，图 5-20(b)为采用罗伯特梯度算子对图像进行锐化处理的结果，图像锐化后仅留下灰度值变化较大的物体边缘点(不同的梯度算子，由于梯度取值方向不同，锐化图像效果略有差异)。



(a) 原始图像



(b) 锐化图像

图 5-20　罗伯特梯度算子图像锐化结果

2. 拉普拉斯算子

拉普拉斯算子是常用的边缘增强算子之一，它也是采用偏导数运算，与梯度法略有不同之处在于拉普拉斯算子采用的是二阶偏导数，其定义为

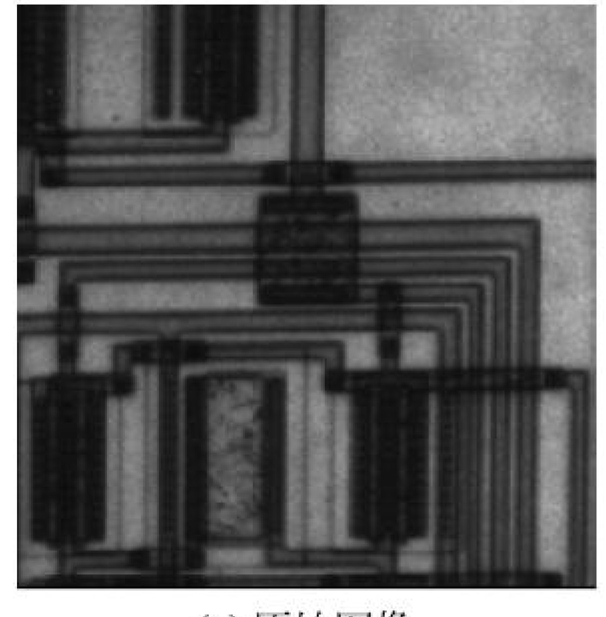
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{5-36}$$

对于数字图像，在某个像素点(x,y)处的拉普拉斯算子可采用如下差分形式近似：

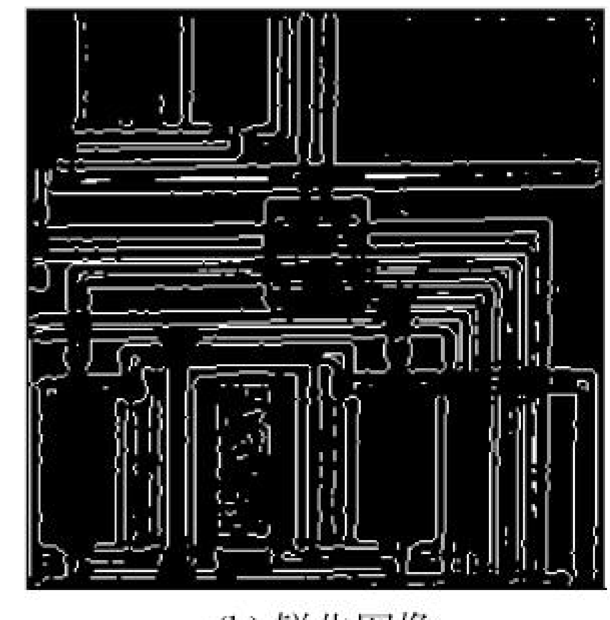
$$\nabla^2 f(x,y) = -f(i,j+1) - f(i,j-1) - f(i+1,j) - f(i-1,j) + 4f(i,j) \tag{5-37}$$

通过罗伯特梯度算子和拉普拉斯算子等典型锐化算法可以看出，平滑模板与锐化模板的不同之处在于，平滑模板各系数的符号均为正，因此平滑具有积分或求和的性质，而锐化模板各系数的符号则有正有负，而且模板系数的和正好为 0，即锐化滤波算法具有微分或差分的性质。数字图像处理中，可以根据应用的需要以及平滑模板与锐化模板的性质，自行设计出满足各种应用需求的模板算子。

如图 5-21 所示是图像锐化效果的又一示例。图 5-21(a)为原始图像，图 5-21(b)为采用拉普拉斯算子对图像进行锐化处理的结果图像，经锐化处理后，图像同样仅留下灰度值变化较大的物体边缘点。



(a) 原始图像



(b) 锐化图像

图 5-21　拉普拉斯算子图像锐化结果

3. 定向滤波

定向滤波是一种定向锐化模板，它是一种对特定方向的物体形迹的增强手段，如山脉与河流的走向等。如图 5-22 所示，分别为水平方向、对角方向和垂直方向的定向滤波模板算子。因此，所谓方向模板是一个各模板系数的大小与符号表现出一定规律性和方向性，并对某些方向的灰度值变化敏感的矩阵，体现在模板卷积的结果上就是在特定方向上取得较大的结果(数值)。

(a) 水平方向

−1	−1	−1
2	2	2
−1	−1	−1

(b) 对角方向

−1	−1	2
−1	2	−1
2	−1	−1

(c) 垂直方向

−1	2	−1
−1	2	−1
−1	2	−1

图 5-22 常用定向滤波模板算子

5.7 空域平滑滤波处理实例

5.7.1 程序源代码

```
%基于空域平滑滤波的消噪处理程序 MyDenoiseSoft
function MyDenoiseSoft(action, varargin)
if nargin<1,
  action = 'InitializeMyDenoiseSoft';
end;
feval(action,varargin{:})
return;
%程序运行初始化函数
%如果本程序已经运行,使之成为当前运行程序
function InitializeMyDenoiseSoft()
h = findobj(allchild(0), 'tag', '基于空域平滑滤波的数字图像消噪处理程序');
if ~isempty(h)
  figure(h(1))
  return
end
screenD = get(0, 'ScreenDepth');
if screenD > 8
  grayres = 256;
else
  grayres = 128;
end
MyDenoiseSoftFig = figure( ...
  'Name','基于空域平滑滤波的数字图像消噪处理程序', ...
  'NumberTitle','off', 'HandleVisibility', 'on', ...
  'tag', '基于空域平滑滤波的数字图像消噪处理程序', ...
  'Visible','off', 'Resize', 'off',...
  'BusyAction','Queue','Interruptible','off', ...
  'Color', [0.8 0.8 0.8], ...
```

```
  'IntegerHandle', 'off', ...
  'DoubleBuffer', 'on', ...
  'Colormap', gray(grayres));
figpos = get(MyDenoiseSoftFig, 'position');
figpos(3:4) = [560 430];
% 调整图像窗口大小
horizDecorations = 10;
vertDecorations = 45;
screenSize = get(0,'ScreenSize');
if (screenSize(3)<=1)
  % No display connected (apparently)
  screenSize(3:4) = [100000 100000];
end
if (((figpos(3) + horizDecorations)> screenSize(3)) | ...
       ((figpos(4) + vertDecorations)> screenSize(4)))
  % 屏幕尺寸太小
  delete(fig);
  error(['Screen resolution is too low ', ...
          '(or text fonts are too big) to run this DenoiseSoft']);
end
dx = screenSize(3) - figpos(1) - figpos(3) - horizDecorations;
dy = screenSize(4) - figpos(2) - figpos(4) - vertDecorations;
if (dx<0)
   figpos(1) = max(5,figpos(1) + dx);
end
if (dy<0)
   figpos(2) = max(5,figpos(2) + dy);
end
set(MyDenoiseSoftFig, 'position', figpos);
rows = figpos(4); cols = figpos(3);
bgcolor = [0.45 0.45 0.45];                              % 背景颜色
wdcolor = [0.8 0.8 0.8];                                 % 窗口颜色
gcolor = [1 1 1];                                        % 文本颜色
hs = (cols - (3 * 128))/4;                               % 水平距离
vs = hs/2;                                               % 垂直距离
ifs = hs/3;                                              % 内部框间距
% 按钮及菜单参数
Std.Interruptible = 'off';
Std.BusyAction = 'queue';
% 图像坐标的默认设置
Ax = Std;
Ax.Units = 'Pixels';
Ax.Parent = MyDenoiseSoftFig;
Ax.ydir = 'reverse';
Ax.XLim = [.5 128.5];
Ax.YLim = [.5 128.5];
Ax.CLim = [0 1];
Ax.XTick = [];
Ax.YTick = [];
Img = Std;
Img.CData = [];
```

```
Img.Xdata = [1 128];
Img.Ydata = [1 128];
Img.CDataMapping = 'Scaled';
Img.Erasemode = 'none';
Ctl = Std;
Ctl.Units = 'Pixels';
Ctl.Parent = MyDenoiseSoftFig;
Btn = Ctl;
Btn.Style = 'pushbutton';
Btn.Enable = 'off';
Edit = Ctl;
Edit.Style = 'edit';
Edit.HorizontalAlignment = 'right';
Edit.BackgroundColor = 'white';
Edit.ForegroundColor = 'black';
Menu = Ctl;
Menu.Style = 'Popupmenu';
Text = Ctl;
Text.Style = 'text';
Text.HorizontalAlignment = 'left';
Text.BackgroundColor = bgcolor;
Text.ForegroundColor = fgcolor;
btnHt = 26;
txtHt = 18;
menuHt = 26;
editHt = 21;
% 原始图像
ud.hOriginalAxes = axes(Ax, ...
   'Position', [hs rows - vs * 1.3 - 128 128 128]);
title('原始图像');
ud.hOriginalImage = image(Img, ...
   'Parent', ud.hOriginalAxes);
%噪声图像
ud.hCorruptedAxes = axes(Ax, ...
   'Position', [2 * hs + 128 rows - vs * 1.3 - 128 128 128]);
title('噪声图像');
ud.hCorruptedImage = image(Img, ...
   'Parent', ud.hCorruptedAxes);
ud.CorruptedImageIsStale = 1;
ud.hFilteredAxes = axes(Ax, ...
   'Position', [3 * hs + 256 rows - vs * 1.3 - 128 128 128]);
title('消噪图像');
ud.hFilteredImage = image(Img, ...
   'Parent', ud.hFilteredAxes);
%图的框架
fleft = ifs;
fbot = btnHt + 3.2 * ifs;
fwid = cols - 2 * hs/3;
fht = rows - vs - hs/2 - 128 - btnHt - 3.2 * ifs;
ud.hControlFrame = uicontrol(Std, ...
   'Parent', MyDenoiseSoftFig, ...
```

```
    'Style', 'Frame', ...
    'Units', 'pixels', ...
    'Position', [fleft fbot fwid fht], ...
    'BackgroundColor', bgcolor);
menuWid = (fwid - 4 * ifs)/3;
menuBot = fbot + fht - vs - menuHt;
labelBot = fbot + fht - 22;
labelWid = menuWid/2;
%图像选择的下拉菜单
ipleft = fleft + ifs;
ud.hImgPop = uicontrol(Menu, ...
    'Position',[ipleft menuBot menuWid menuHt], ...
    'String','辣椒|钱币 1|电路图|米饭图|嘛嘴|土星|钱币 2|玻璃|漂浮物', ...
    'Callback','MyDenoiseSoft(''LoadNewImage'')');
%图像菜单的提示文本
uicontrol( Text, ...
    'Position',[ipleft labelBot menuWid txtHt], ...
    'String','选择原始图像:');
%添加噪声菜单
anleft = fleft + 2 * ifs + menuWid;
ud.hAddNoisePop = uicontrol(Menu, ...
    'Position',[anleft menuBot menuWid menuHt], ...
    'String','胡椒加盐|高斯噪声|斑点噪声', ...
    'Callback','MyDenoiseSoft(''UpdateNoiseType'')');
%添加噪声菜单的提示文本
uicontrol( Text, ...
    'Position',[anleft labelBot menuWid txtHt], ...
    'String','噪声类型选择:');
%滤波方法选择菜单
nrleft = fleft + 3 * ifs + 2 * menuWid;
ud.hNoiseRemovalPop = uicontrol(Menu, ...
    'Position',[nrleft menuBot menuWid menuHt], ...
    'String','中值滤波|均值滤波|自适应滤波', ...
    'Callback','MyDenoiseSoft(''UpdateFilterType'')');
%滤波方法文本提示
uicontrol( Text, ...
    'Position',[nrleft labelBot menuWid txtHt], ...
    'String','噪声消除方法选择:');
%高斯噪声
ud.hGaussianMean = uicontrol(Edit, ...
    'Position',[anleft + labelWid menuBot - editHt - ifs menuWid - labelWid editHt], ...
    'String','0.0',...
    'callback','MyDenoiseSoft(''EditBoxUpdate'',''GaussianMean'')');
ud.hTxtGaussianMean = uicontrol( Text, ...
    'Position',[anleft menuBot - editHt - ifs labelWid txtHt], ...
    'String','Mean:');
ud.GaussianMean = 0.0;
ud.hGaussianVariance = uicontrol(Edit, ...
    'Position',[anleft + labelWid menuBot - 2 * editHt - 2 * ifs menuWid - labelWid editHt], ...
    'String','0.005',...
    'callback','MyDenoiseSoft(''EditBoxUpdate'',''GaussianVariance'')');
```

```
ud.hTxtGaussianVariance = uicontrol( Text, ...
    'Position',[anleft menuBot - 2 * editHt - 2 * ifs labelWid txtHt], ...
    'String','高斯噪声方差:');
ud.GaussianVariance  = 0.005;
% 胡椒加盐噪声密度
ud.hSaltNPepperDensity = uicontrol(Edit, ...
    'Position',[anleft + labelWid menuBot - editHt - ifs menuWid - labelWid editHt], ...
    'String','0.1',...
    'callback','MyDenoiseSoft(''EditBoxUpdate'',''SaltNPepperDensity'')');
ud.hTxtSaltNPepperDensity = uicontrol( Text, ...
    'Position',[anleft menuBot - editHt - ifs labelWid txtHt], ...
    'String','噪声密度:');
ud.SaltNPepperDensity = 0.1;
% 斑点噪声变量
ud.hSpeckleVariance = uicontrol(Edit, ...
    'Position',[anleft + labelWid menuBot - editHt - ifs menuWid - labelWid editHt], ...
    'String','0.04',...
    'callback','MyDenoiseSoft(''EditBoxUpdate'',''SpeckleVariance'')');
ud.hTxtSpeckleVariance = uicontrol( Text, ...
    'Position',[anleft menuBot - editHt - ifs labelWid txtHt], ...
    'String','斑点噪声方差:');
ud.SpeckleVariance = 0.04;
% "添加噪声"按钮
ud.hAddNoiseBtn = uicontrol(Btn, ...
    'Position',[anleft fbot + ifs menuWid btnHt], ...
    'String','添加噪声', ...
    'Callback','MyDenoiseSoft(''AddNoise'')');
% 中值滤波模板选择
ud.hNeighborhoodPop = uicontrol(Menu, ...
    'Position',[nrleft menuBot - menuHt - ifs - txtHt menuWid menuHt], ...
    'String','3 × 3|5 × 5|7 × 7', ...
    'Callback','MyDenoiseSoft(''UpdateNeighborhoodSize'')');
% 图像菜单的文本标识
uicontrol( Text, ...
    'Position',[nrleft menuBot - ifs - txtHt menuWid txtHt], ...
    'String','模板选择:');
% "滤波处理"功能按钮
ud.hApplyFilterBtn = uicontrol(Btn, ...
    'Position',[nrleft fbot + ifs menuWid btnHt], ...
    'String','滤波处理', ...
    'Callback','MyDenoiseSoft(''ApplyFilter'')');
% "信息"显示与"退出"功能
spac = ifs * 2/3;
ficleft = cols/2;
ficbot = ifs;
ficht = btnHt + 2 * spac;
ficwid = cols/2 - ifs;
ud.hInfoCloseFrame = uicontrol(Std, ...
    'Parent', MyDenoiseSoftFig, ...
    'Style', 'Frame', ...
    'Units', 'pixels', ...
```

```
    'Position', [ficleft ficbot ficwid ficht], ...
    'BackgroundColor', bgcolor);
%"信息"显示与"退出"功能按钮
ud.hInfo = uicontrol(Btn, ...
    'Position',[ficleft + spac ficbot + spac (ficwid - 3 * spac)/2 btnHt], ...
    'String','信息', ...
    'Callback','helpwin MyDenoiseSoft');
ud.hClose = uicontrol(Btn, ...
    'Position',[ficleft + ficwid/2 + spac/2 ficbot + spac (ficwid - 3 * spac)/2 btnHt], ...
    'String','退出', ...
    'Callback','close(gcbf)');
%状态条
ud.hStatus = uicontrol(Std, ...
    'Parent', MyDenoiseSoftFig, ...
    'Style','text', ...
    'Units','pixels', ...
    'Position',[ifs ifs cols/2 - 2 * ifs txtHt * 2], ...
    'Foreground', [.8 0 0], ...
    'Background',wdcolor, ...
    'Horiz','center', ...
    'Tag', 'Status', ...
    'String','Initializing MyDenoiseSoft...');
set(MyDenoiseSoftFig, 'UserData', ud);
UpdateNoiseType(MyDenoiseSoftFig);
UpdateFilterType(MyDenoiseSoftFig);
UpdateNeighborhoodSize(MyDenoiseSoftFig);
set(MyDenoiseSoftFig, 'visible','on','HandleVisibility','callback');

LoadNewImage(MyDenoiseSoftFig)
set([ud.hAddNoiseBtn ud.hApplyFilterBtn ud.hInfo ud.hClose], 'Enable', 'on');
return
%添加噪声到原始图像
function AddNoise(MyNoiseFig)
if nargin < 1
   MyNoiseFig = gcbf;
end
set(MyNoiseFig,'Pointer','watch');
setstatus(MyNoiseFig,'adding noise to the original image...');
ud = get(MyNoiseFig,'Userdata');
orig = getimage(ud.hOriginalImage);
switch ud.NoiseType
case '胡椒加盐'
   noisy = imnoise(orig, 'Salt & Pepper', ud.SaltNPepperDensity);
case '高斯噪声'
   noisy = imnoise(orig, 'Gaussian', ud.GaussianMean, ud.GaussianVariance);
case '斑点噪声'
   noisy = imnoise(orig, 'Speckle', ud.SpeckleVariance);
end
set(ud.hCorruptedImage, 'Cdata', noisy);
AddNoiseOff_ApplyFilterOn(MyNoiseFig);
set(MyNoiseFig,'Pointer','arrow');
```

```
setstatus(MyNoiseFig,'按"滤波处理" 按钮消除噪声.');
ud.CorruptedImageIsStale = 0;
set(MyNoiseFig, 'UserData', ud);
drawnow
return;
%对噪声图像进行滤波
function ApplyFilter(MyNoiseFig)
if nargin<1
   MyNoiseFig = gcbf;
end
set(MyNoiseFig,'Pointer','watch');
setstatus(MyNoiseFig,'Applying noise removal filter to the corrupted image...');
ud = get(MyNoiseFig,'Userdata');
noisy = getimage(ud.hCorruptedImage);
hood = ud.NeighborhoodSize;
switch ud.FilterType
case '中值滤波'
   filtered = medfilt2(noisy, [hood hood]);
case '均值滤波'
   myfilt = ones(hood)/(hood^2);                    % Boxcar filter
   filtered = filter2(myfilt,noisy,'same');
case '自适应滤波'
   filtered = wiener2(noisy,[hood hood]);
end
set(ud.hFilteredImage, 'Cdata', filtered);
setstatus(MyNoiseFig,'');
set(ud.hApplyFilterBtn, 'enable', 'off');
set(MyNoiseFig,'Pointer','arrow');
drawnow
return;
%从图像库中读新图像
function LoadNewImage(MyNoiseFig)
if nargin<1
   MyNoiseFig = gcbf;
end
set(MyNoiseFig,'Pointer','watch');
ud = get(MyNoiseFig,'Userdata');
v = get(ud.hImgPop,{'value','String'});
name = deblank(v{2}(v{1},:));
drawnow
switch name
case '钱币 1'
   coins = [];
   load imdemos coins %生成新变量
   img = coins;
case '土星',
   saturn = [];
   load imdemos saturn
   img = saturn;
case '噘嘴',
   pout = [];
```

```
        load imdemos pout
        img = pout;
    case '钱币 2',
        quarter = [];
        load imdemos quarter
        img = quarter;
    case '电路图',
        circuit = [];
        load imdemos circuit
        img = circuit;
    case '漂浮物',
        liftbody128 = [];
        load imdemos liftbody128
        img = liftbody128;
    case '辣椒',
        pepper = [];
        load imdemos pepper
        img = pepper;
    case '米饭图',
        rice = [];
        load imdemos rice
        img = rice;
    case '玻璃',
        glass = [];
        load imdemos glass
        img = glass;
    otherwise
        error('MyDenoiseSoft: Unknown Image Option!');
    end
    img = double(img)/255;
    set(ud.hOriginalImage, 'Cdata', img);
    set(ud.hAddNoiseBtn, 'Enable', 'on');
    AddNoise(MyNoiseFig);
    ApplyFilter(MyNoiseFig);
    return;
    %更新编框用户变量值
    function EditBoxUpdate(whichBox)
    MyNoiseFig = gcbf;
    ud = get(MyNoiseFig, 'Userdata');
    switch whichBox
    case 'SpeckleVariance'
        str = get(ud.hSpeckleVariance, 'String');
        new = str2double(str);
        if isempty(new) | prod(size(new))>1 | new>1 | new<0
            errordlg('斑点噪声方差值应在[0,1]之间');
        else
            ud.SpeckleVariance = real(new(1));
        end
        set(ud.hSpeckleVariance, 'String', num2str(ud.SpeckleVariance));
    case 'SaltNPepperDensity'
        str = get(ud.hSaltNPepperDensity, 'String');
```

```
    new = str2double(str);
    if isempty(new) | prod(size(new))>1 | abs(new)>1
        errordlg('胡椒加盐噪声方差值应在[0,1]之间');
    else
        ud.SaltNPepperDensity = real(new(1));
    end
    set(ud.hSaltNPepperDensity, 'String', num2str(ud.SaltNPepperDensity));
case 'GaussianVariance'
    str = get(ud.hGaussianVariance, 'String');
    new = str2double(str);
    if isempty(new) | prod(size(new))>1 | new > 1 | new < 0
        errordlg('高斯噪声方差值应在[0,1]之间');
    else
        ud.GaussianVariance = real(new(1));
    end
    set(ud.hGaussianVariance, 'String', num2str(ud.GaussianVariance))
case 'GaussianMean'
    str = get(ud.hGaussianMean, 'String');
    new = str2double(str);
    if isempty(new) | prod(size(new))>1 | abs(new)>1
        errordlg('The mean for Gaussian noise should be a scalar in [-1,1]');
    else
        ud.GaussianMean = real(new(1));
    end
    set(ud.hGaussianMean, 'String', num2str(ud.GaussianMean))
end
AddNoiseOn_ApplyFilterOff(MyNoiseFig);
ud.CorruptedImageIsStale = 1;
setstatus('按"添加噪声" 按钮给原始图像添加噪声..');
set(MyNoiseFig,'Userdata',ud);
%"滤波处理"的允许与"添加噪声"按钮的禁用功能的实现函数
function AddNoiseOff_ApplyFilterOn(MyNoiseFig)
ud = get(MyNoiseFig, 'UserData');
set(ud.hAddNoiseBtn, 'enable', 'off');
set(ud.hApplyFilterBtn, 'enable', 'on');

%"滤波处理"的禁用与"添加噪声"按钮的允许功能的实现函数
function AddNoiseOn_ApplyFilterOff(MyNoiseFig)
ud = get(MyNoiseFig, 'UserData');
set(ud.hAddNoiseBtn, 'enable', 'on');
set(ud.hApplyFilterBtn, 'enable', 'off');
return
%噪声类型更新
function UpdateNoiseType(MyNoiseFig)
if nargin<1
    MyNoiseFig = gcbf;
end
set(MyNoiseFig,'Pointer','watch');
ud = get(MyNoiseFig,'Userdata');
v = get(ud.hAddNoisePop,{'value','String'});
noisetype = deblank(v{2}(v{1},:));
```

```
ud.NoiseType = noisetype;
drawnow
switch noisetype
case 'Salt & Pepper'
   BringUpSaltNPepperControls(MyNoiseFig);
case 'Gaussian'
   BringUpGaussianControls(MyNoiseFig);
case 'Speckle'
   BringUpSpeckleControls(MyNoiseFig);
end
AddNoiseOn_ApplyFilterOff(MyNoiseFig);
setstatus('按"添加噪声" 按钮添加噪声.');
ud.CorruptedImageIsStale = 1;
set(MyNoiseFig,'Pointer','arrow','Userdata', ud);
return
%实现高斯噪声(产生高斯噪声)
function BringUpGaussianControls(MyNoiseFig)
if nargin < 1
   MyNoiseFig = gcbf;
end
ud = get(MyNoiseFig,'Userdata');
%胡椒加盐
set([ud.hSaltNPepperDensity ud.hTxtSaltNPepperDensity], ...
   'Enable', 'off', 'visible', 'off');
%斑点
set([ud.hSpeckleVariance ud.hTxtSpeckleVariance], ...
   'Enable', 'off', 'visible', 'off');
%高斯
set([ud.hGaussianMean ud.hTxtGaussianMean ud.hGaussianVariance ...
    ud.hTxtGaussianVariance], 'Enable', 'on', 'visible', 'on');
return
%实现胡椒加盐噪声(产生胡椒加盐噪声)
function BringUpSaltNPepperControls(MyNoiseFig)
if nargin < 1
  MyNoiseFig = gcbf;
end
ud = get(MyNoiseFig,'Userdata');
%高斯
set([ud.hGaussianMean ud.hTxtGaussianMean ud.hGaussianVariance ...
    ud.hTxtGaussianVariance], 'Enable', 'off', 'visible', 'off');
%斑点
set([ud.hSpeckleVariance ud.hTxtSpeckleVariance], ...
   'Enable', 'off', 'visible', 'off');
%胡椒加盐
set([ud.hSaltNPepperDensity ud.hTxtSaltNPepperDensity], ...
   'Enable', 'on', 'visible', 'on');
%实现斑点噪声
function BringUpSpeckleControls(MyNoiseFig)
if nargin < 1
  MyNoiseFig = gcbf;
end
```

```
ud = get(MyNoiseFig, 'Userdata');
% 高斯
set([ud.hGaussianMean ud.hTxtGaussianMean ud.hGaussianVariance ...
      ud.hTxtGaussianVariance], 'Enable', 'off', 'visible', 'off');
% 胡椒加盐
set([ud.hSaltNPepperDensity ud.hTxtSaltNPepperDensity], ...
   'Enable', 'off', 'visible', 'off');
% 斑点
set([ud.hSpeckleVariance ud.hTxtSpeckleVariance], ...
   'Enable', 'on', 'visible', 'on');
% 更新滤波方法
function UpdateFilterType(MyNoiseFig)
if nargin < 1
  MyNoiseFig = gcbf;
end
ud = get(MyNoiseFig, 'Userdata');
v = get(ud.hNoiseRemovalPop, {'value', 'String'});
ud.FilterType =  deblank(v{2}(v{1},:));
if ~ud.CorruptedImageIsStale
  set(ud.hApplyFilterBtn, 'Enable', 'on');
  setstatus('按"滤波处理" 按钮消除噪声.');
end
set(MyNoiseFig, 'UserData', ud);

% 更新邻域模板
function UpdateNeighborhoodSize(MyNoiseFig)
if nargin < 1
  MyNoiseFig = gcbf;
end
set(MyNoiseFig, 'Pointer', 'watch');
ud = get(MyNoiseFig, 'Userdata');
v = get(ud.hNeighborhoodPop, {'value', 'String'});
neighborhood = deblank(v{2}(v{1},:));
ud.NeighborhoodSize = str2double(neighborhood(1));
if ~ud.CorruptedImageIsStale
  set(ud.hApplyFilterBtn, 'Enable', 'on');
  setstatus('按"滤波处理" 按钮消除噪声.');
end
set(MyNoiseFig, 'UserData', ud, 'Pointer', 'arrow');
```

5.7.2 应用举例

程序 MyDenoiseSoft 建立在 MATLAB 2008 软件包的基础上，在 MATLAB 环境下运行该程序，可进行基于空域平滑滤波的数字图像消噪处理运算方面的应用，该程序可进行不同噪声条件下的中值滤波、均值滤波、自适应滤波等方面的消除噪声应用。程序运行界面如图 5-23 所示。

如图 5-24 所示，单击“选择原始图像”下方的三角形按钮，可选择已经给定的各种原始图像，单击“噪声类型选择”可对原始图像添加胡椒加盐等三种具有典型意义的噪声类型。本程序还包括“抑制噪声方法选择”等对话框，从中可选择中值滤波、均值滤波和自适应滤波

等消噪方法。



图 5-23 程序 MyDenoiseSoft 运行结果



(a) 选择原始图像



(b) 噪声类型选择

图 5-24 程序 MyDenoiseSoft 部分对话框

习题

1. 图像噪声按其产生的原因可以分为几种类型?
2. 图像噪声具有哪些特点?
3. 图像增强技术能增加原始图像的信息吗? 为什么?
4. 简述图像增强方法的主要类别。
5. 图像增强效果的评价方法主要包括哪些?
6. 在图像增强技术中,常用的空域灰度变换包括哪几种?
7. 什么是线性灰度拉伸? 线性灰度拉伸可以分为几种情况?
8. 非线性拉伸主要包括哪些类型? 试比较非线性拉伸与分段线性拉伸各有哪些特点。
9. 试简述直方图均衡化实现图像增强的主要步骤。
10. 通常情况下,对数字图像的直方图均衡化可以产生完全均匀分布的直方图吗? 为

什么？

11. 什么是直方图均衡化的“简并”现象？通常情况下，被简并的是原始图像中的哪些信息成分？

12. 选定一幅图像，编写 MATLAB 程序对其实现直方图均衡化处理，根据处理结果分析均衡化对图像有哪些作用。

13. 什么是直方图规定化技术？简述直方图规定化图像增强的主要步骤。

14. 试总结和比较直方图规定化与直方图均衡化的实现原理。

15. 若一幅图像共有 8 个灰度级，每一灰度级的概率分布如表 5-7 所示，现要求根据如表 5-8 所示的规定化直方图数据，对原始图像进行直方图规定化处理。

表 5-7 原始图像灰度分布表

r_k	$P_r(r_k)=\frac{n_k}{n}$
$r_0=0$	0.174
$r_1=1/7$	0.088
$r_2=2/7$	0.086
$r_3=3/7$	0.080
$r_4=4/7$	0.068
$r_5=5/7$	0.058
$r_6=6/7$	0.062
$r_7=1$	0.384

表 5-8 规定化直方图灰度分布表

z_k	$P_z(z_k)$
$z_0=0$	0.00
$z_1=1/7$	0.40
$z_2=2/7$	0.00
$z_3=3/7$	0.00
$z_4=4/7$	0.20
$z_5=5/7$	0.00
$z_6=6/7$	0.00
$z_7=1$	0.4

16. 数字图像的空域滤波主要包括哪几种情况？

17. 什么是邻域平均滤波法？它具有哪些主要特点？

18. 选择一幅图像，编写 MATLAB 程序对其进行邻域平均滤波，观察滤波效果，总结邻域平均滤波的适应性和相关特点。

19. 非线性滤波中常用的方法有哪几种？各有何特点？

20. 选择一幅图像并添加椒盐噪声，编写 MATLAB 程序对图像进行中值滤波，观察滤波效果。

21. 什么是锐化滤波？常用的锐化滤波方法主要有哪几种？

第 6 章 图像频域增强

CHAPTER 6

图像增强是数字图像处理常用的技术之一,其目的是改善图像的质量以达到突出图像特征或达到赏心悦目的效果。数字图像处理方法主要分为两大类:一类是在空间域直接对图像进行处理,称为空域法;另一类是变换到某种变换域对图像进行分析和处理,称为变换域处理法。基于傅里叶变换的频域分析方法物理意义明晰,是一种非常重要的变换域处理法,在图像处理中具有广泛的应用。本章主要介绍图像的频域增强方法,这些方法包括频域低通滤波、高通滤波、带通滤波、带阻滤波和同态滤波等。

6.1 低通滤波

图像从空间域变换到频率域后,其低频分量对应图像中灰度值变化比较缓慢的区域,而高频分量则表示了图像中物体的边缘和随机噪声等信息。低通滤波器的功能是通过低通滤波器函数 H 减弱或抑制高频分量,保留低频分量。因此,低通滤波与空域中的平滑滤波器在功能上一样,可以消除图像中的随机噪声、削弱边缘效应,起到平滑图像的作用。

傅里叶变换和卷积理论是频域滤波技术的基础,因此,在频率域空间的滤波与空域滤波一样可以通过卷积运算实现。常用的低通滤波器包括理想低通滤波器、巴特沃斯低通滤波器、指数低通滤波器和梯形低通滤波器等多种类型。

设 $g(x,y)$表示函数 $f(x,y)$与线性移不变算子 $h(x,y)$进行卷积运算的结果,即

$$g(x,y) = f(x,y) * h(x,y) \tag{6-1}$$

因此可得

$$G(u,v) = F(u,v)H(u,v) \tag{6-2}$$

式中,G、F、H 分别是函数 $g(x,y)$、$f(x,y)$、$h(x,y)$的傅里叶变换;$H(u,v)$称为滤波器函数,也可以称为传递函数。在图像增强中,由于待增强的图像函数 $f(x,y)$是已知的,因此 $F(u,v)$可由图像的傅里叶变换得到。

实际应用中,首先需要确定 $H(u,v)$,然后就可以求得 $G(u,v)$,再对 $G(u,v)$进行傅里叶逆变换,即可得到增强的图像 $g(x,y)$。$g(x,y)$可以突出 $f(x,y)$的某一方面的特征信息。若通过 $H(u,v)$增强 $F(u,v)$的高频信息,如增强图像的边缘信息等,则为高通滤波;如果增强 $F(u,v)$的低频信息,如对图像进行平滑操作等,则为低通滤波。频域滤波方法的系统框图如图 6-1 所示。其滤波处理过程可以分为以下三个步骤:

（1）对原始图像 $f(x,y)$进行傅里叶变换得到 $F(u,v)$。

（2）将 $F(u,v)$与滤波器函数 $H(u,v)$进行卷积运算得到 $G(u,v)$。

（3）对 $G(u,v)$进行傅里叶逆变换，即可求出增强图像 $g(x,y)$。

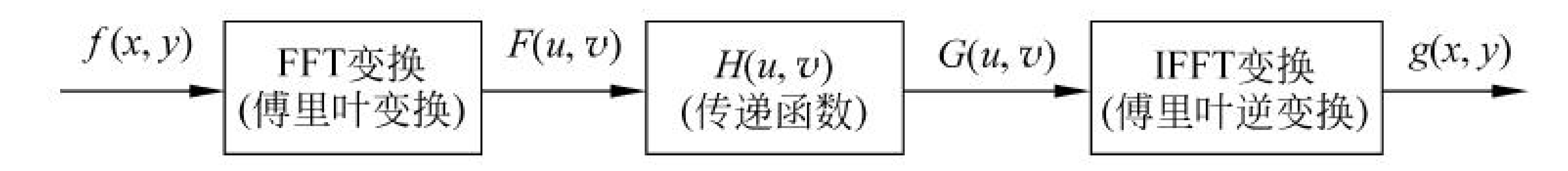


图 6-1　频域滤波系统框图

6.1.1　理想低通滤波器

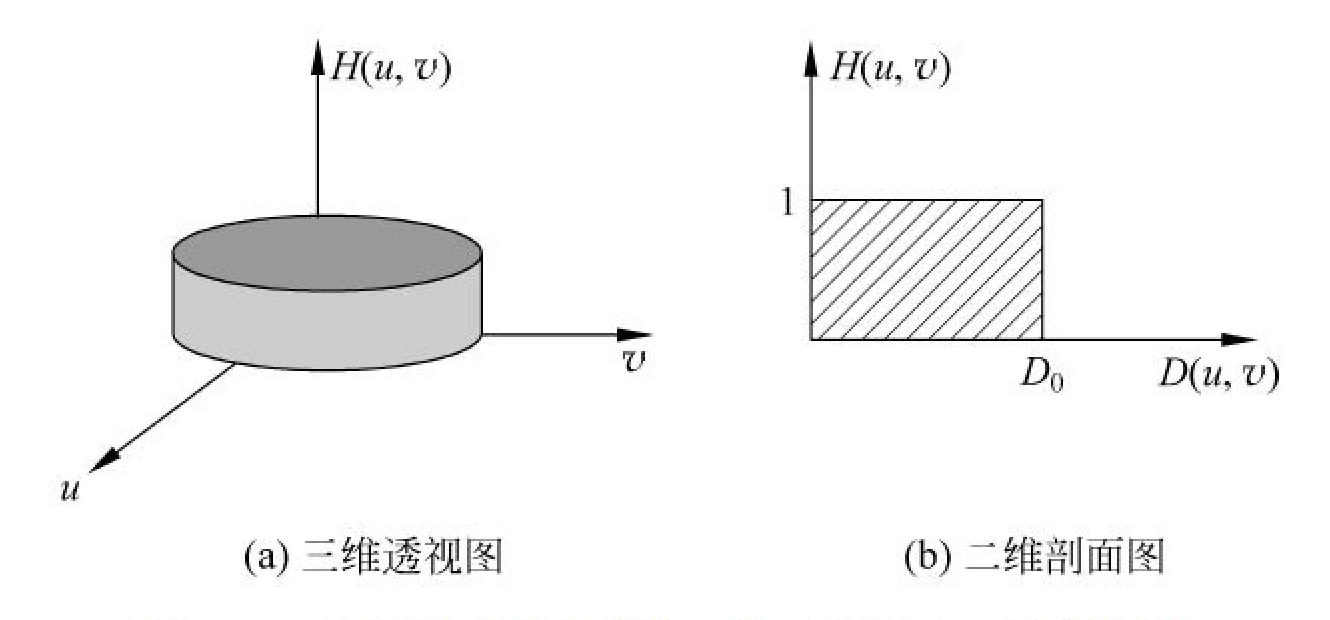
本章只讨论径向对称的零相移滤波器函数，几种常用的低通滤波器形式如下。二维的理想低通滤波器的传递函数为

$$H(u,v)=\begin{cases}1 & D(u,v)\leqslant D_0\\0 & D(u,v)>D_0\end{cases}\tag{6-3}$$

式中，D_0 是一个非负整数，即理想低通滤波器的截止频率；$D(u,v)$是从点(u,v)到频域原点的距离，即

$$D(u,v)=\sqrt{u^2+v^2}\tag{6-4}$$

因此，$H(u,v)$、u、v 组成了理想低通滤波器的三维图形。如图 6-2 所示，(a)和(b)分别为理想低通滤波器的三维透视图和二维剖面图。理想低通滤波器的作用是将小于 D_0 的频率，即以 D_0 为半径的圆内的所有频率成分可以无衰减通过，而大于 D_0 的频率则被完全截止不能通过。



(a) 三维透视图　　(b) 二维剖面图

图 6-2　理想低通滤波器的三维透视图和二维剖面图

理论上给出的滤波器函数(包括高通滤波)形式都是以坐标原点径向对称的，而对于一个数字图像所对应的 $N\times N$ 频域矩阵，坐标原点是该矩形的中心，因而滤波器理想特性一般如图 6-2 所示。理想低通滤波器的数学定义形式非常简洁，其平滑作用的物理意义非常明显，但在图像处理过程中会产生比较严重的模糊与振铃现象。因为，根据傅里叶变换的性质，若 $H(u,v)$为理想的矩形特性，那么其逆变换 $h(x,y)$的特性会产生无限的振铃特性，$h(x,y)$与 $f(x,y)$卷积运算后将给目标图像 $g(x,y)$造成模糊与振铃现象。而且 D_0 越小，这种现象越明显。

此外，在截止频率 D_0 处垂直截止的理想低通滤波器只能通过计算机模拟实现，无法采用电子器件实现。

6.1.2　巴特沃斯低通滤波器

巴特沃斯(Butterworth)低通滤波器的传递函数为

$$H(u,v)=\frac{1}{1+\left(\frac{D(u,v)}{D_0}\right)^{2n}} \tag{6-5}$$

式中，D_0 为截止频率；n 为滤波器的阶次。

和理想低通滤波器一样，巴特沃斯低通滤波器的特性曲线同样为三维图形，其剖面示意图如图 6-3 所示。一般情况下，当 $H(u,v)$ 下降至最大值的 1/2 时的 $D(u,v)$ 为截止频率 D_0。实际应用中，有时也取 $H(u,v)$ 下降至最大值的 0.707 时的 $D(u,v)$ 作为截止频率 D_0，这时传递函数为

$$H(u,v)=\frac{1}{1+(\sqrt{2}-1)\left[\frac{D(u,v)}{D_0}\right]^{2n}} \tag{6-6}$$

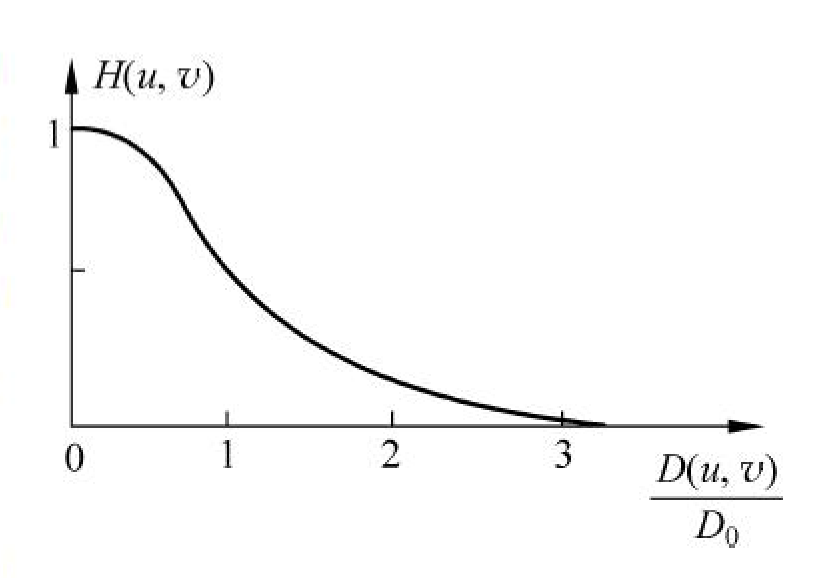


图 6-3　巴特沃斯低通滤波器剖面示意图

巴特沃斯低通滤波器又称为最大平坦滤波器，其通带与阻带之间的过渡比较平坦。因此，巴特沃斯低通滤波器的特点是在通过频率与截止频率之间没有明显的不连续性，不会出现“振铃”现象，其效果好于理想低通滤波器。

例 6-1：数字图像巴特沃斯低通滤波实验

如图 6-4 所示是图像的巴特沃斯低通铝箔实例。图 6-4(a)为带高频噪声的原始图像，图 6-4(b)为对原始图像进行巴特沃斯低通滤波的结果。观察原始图像，显然噪声信号比图像中的电路变化快很多，属于高频信号，经过低通滤波以后，较好地消除了噪声，但图像也一定程度上变模糊了。

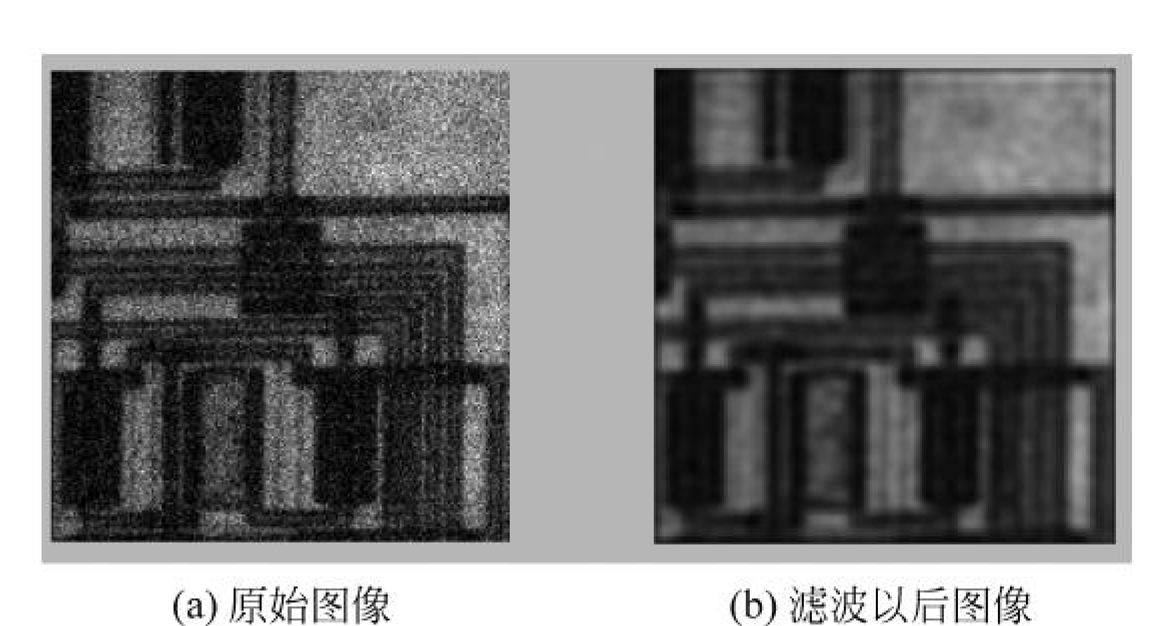

(a) 原始图像　　(b) 滤波以后图像

图 6-4　巴特沃斯低通滤波消除噪声

巴特沃斯低通滤波 MATLAB 代码如下：

```
%巴特沃斯低通滤波
clc;
close all;
image_circuit = imread('circuit.tif');
image_circuit = imnoise(image_circuit,'gaussian',0.05);
I = im2double(image_circuit);
```

```
M = 2 * size(I,1);
N = 2 * size(I,2);
u = - M/2:(M/2 - 1);
v = - N/2:(N/2 - 1);
[U,V] = meshgrid(u,v);
D = sqrt(U.^2 + V.^2);
D0 = 50;
n = 6;
H = 1./(1 + (D./D0).^(2 * n));
F = fftshift(fft2(I,size(H,1),size(H,2)));
G = F. * H;
L = ifft2(fftshift(G));
L = L(1:size(I,1),1:size(I,2));
subplot(121);
imshow(image_circuit);
subplot(122);
imshow(L);
```

6.1.3 指数低通滤波器

指数低通滤波器的传递函数为

$$H(u,v) = e^{-\left[\frac{D(u,v)}{D_0}\right]^n} \tag{6-7}$$

一般情况下，取 $H(u,v)$下降至最大值的 1/2 时的 $D(u,v)$为截止频率 D_0，其剖面图如图 6-5 所示。与巴特沃斯低通滤波器一样，指数低通滤波器从通过频率到截止频率之间具有一段平滑的过渡带，也没有明显的不连续性。

6.1.4 梯形低通滤波器

梯形低通滤波器的传递函数为

$$H(u,v) = \begin{cases} 1 & D(u,v) < D_0 \\ \dfrac{D(u,v) - D_1}{D_0 - D_1} & D_0 \leqslant D(u,v) \leqslant D_1 \\ 0 & D(u,v) > D_1 \end{cases} \tag{6-8}$$

梯形低通滤波器的剖面图如图 6-6 所示。从中可以看出，在 D_0 的尾部包含有一部分高频分量($D_1 > D_0$)，因而，结果图像的清晰度较理想低通滤波器有所改善，振铃效应也有所减弱，应用时通过调整 D_1 值，既能达到平滑图像的目的，又能使图像保持足够的清晰度。

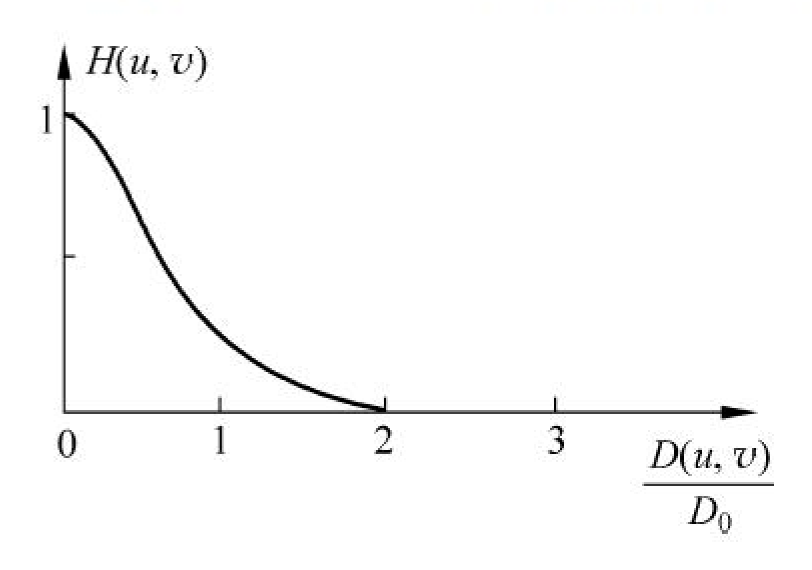


图 6-5 指数低通滤波器剖面图

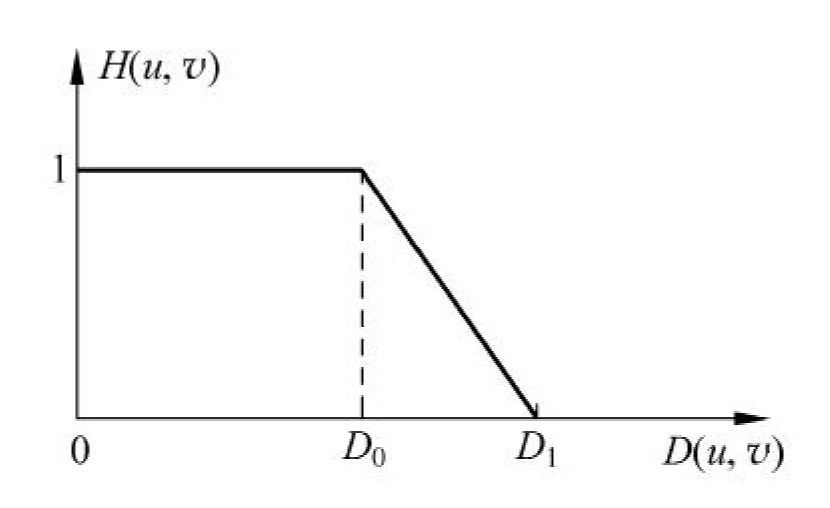


图 6-6 梯形低通滤波器剖面图

根据DFT变换的性质和频谱分布特点，二维DFT的频谱主要集中在低频处。因此，在设计和应用低通滤波器时，一定要注意二维DFT的频谱特点，如例6-2所示。

例6-2：数字图像低通滤波实验

如图6-7所示是图像低通滤波及其截止频率对滤波效果的影响实验。图6-7(a)为一幅256×256的数字图像，图6-7(b)为原始图像的傅里叶频谱。当选取截止频率到原点的距离$D_0=5$时，理想低通滤波器可以保存能量的90%；当$D_0=11$时，通过的能量迅速增加，低通滤波器能保存能量的95%；当$D_0=22$时，则可以保存总能量的98%；当$D_0=45$时，则可以保存总能量的99%。图6-7(c)、图6-7(d)、图6-7(e)、图6-7(f)分别为D_0取上述数值时的滤波效果。因此，合理地选取D_0是应用低通滤波器平滑图像的关键。

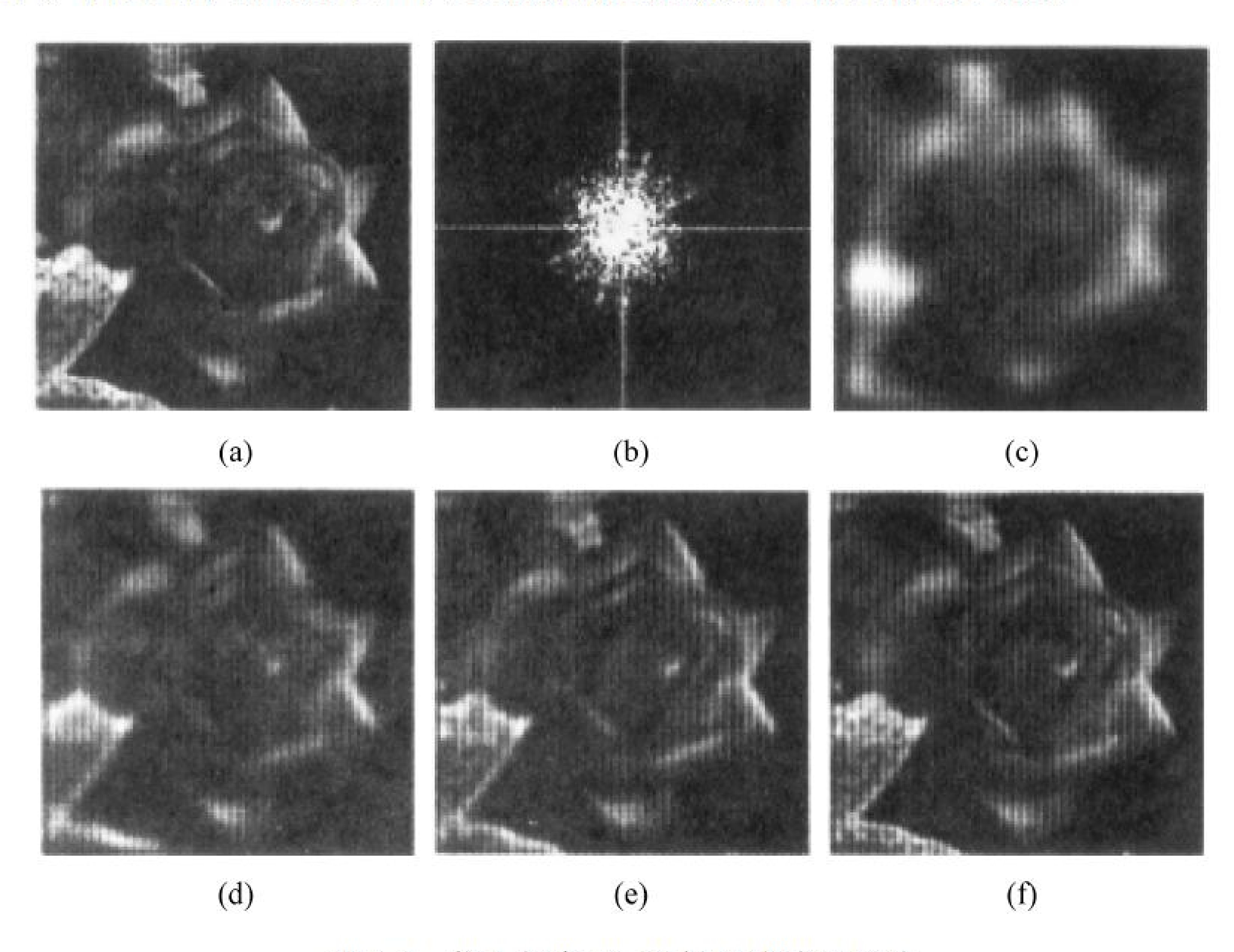


图6-7　截止频率D_0对低通滤波的影响

6.2　高通滤波

图像中物体的边缘及其他灰度变化较快的区域与图像的高频信息有关，因此利用高通滤波器可以对图像的边缘信息进行增强，起到锐化图像的作用。高通滤波器包括理想高通滤波器、巴特沃斯高通滤波器、指数高通滤波器和梯形高通滤波器等多种类型。与低通滤波器相同，本章只讨论径向对称的零相移滤波器函数。

6.2.1　理想高通滤波器

二维理想高通滤波器的传递函数为

$$H(u,v)=\begin{cases}0 & D(u,v)\leqslant D_0\\1 & D(u,v)>D_0\end{cases} \tag{6-9}$$

式中，D_0是一个非负整数，即理想高通滤波器的截止频率；$D(u,v)$是从点(u,v)到频

域原点的距离。

$$D(u,v) = \sqrt{u^2 + v^2}$$

图 6-8 给出了理想高通滤波器滤波特性的剖面图。其作用与理想低通滤波器相反，它将小于 D_0 的频率（半径为 D_0 的圆内）完全截止，而大于 D_0 的频率（圆外的频率）则可以全部无衰减通过。理想高通滤波器也不能通过电子器件实现。

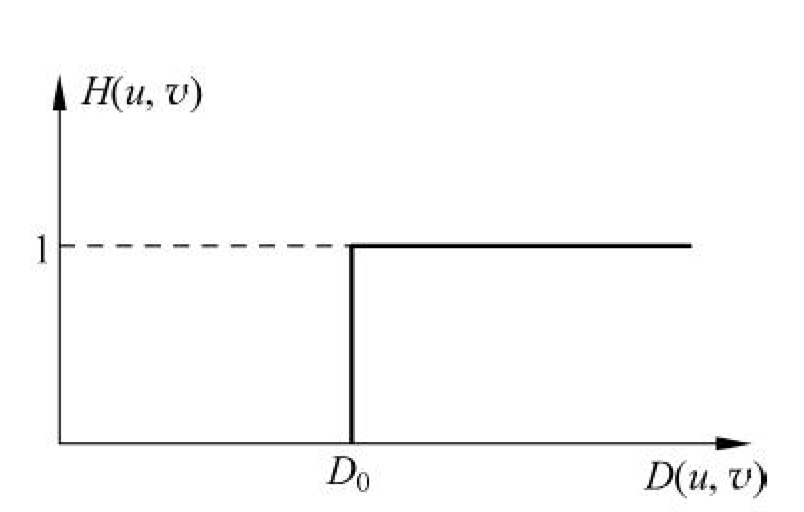


图 6-8 理想高通滤波器剖面图

下例为数字图像的理想高通滤波实验。

例 6-3：数字图像理想高通滤波实验

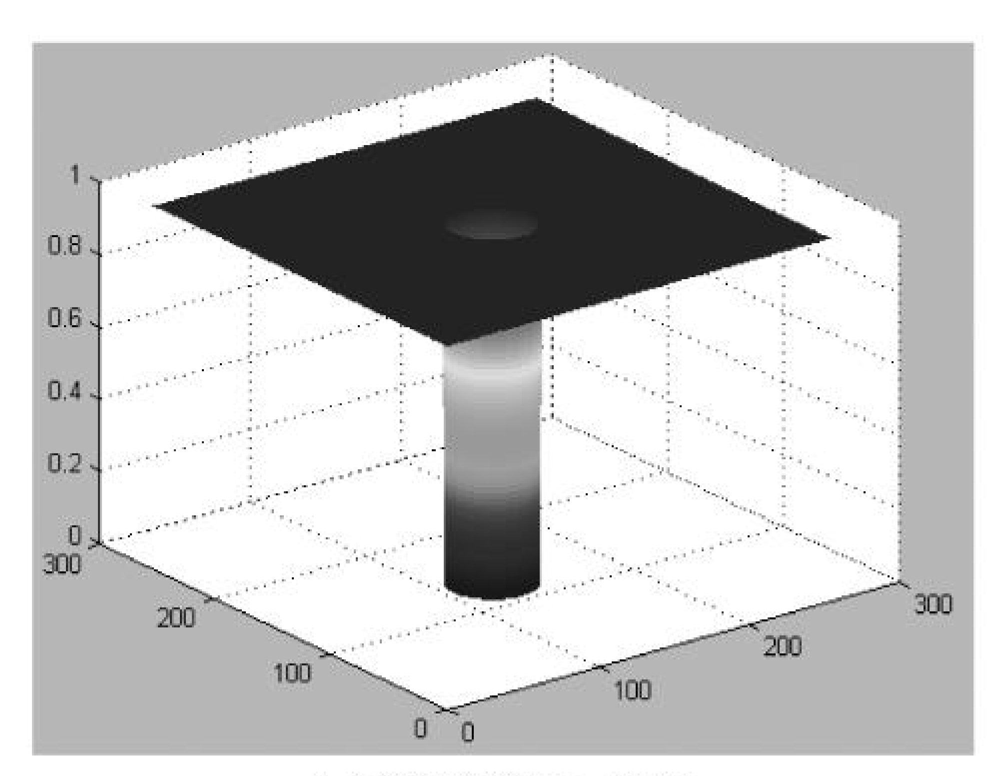
如图 6-9 所示是理想高通滤波消除噪声的实例。图 6-9(a)为原始图像，图 6-9(b)为对原始图像进行理想高通滤波的结果，图 6-9(c)为高通滤波器三维图。由于对原始图像滤除了低频信号，因此，图 6-9(b)仅余下高频边缘信号。半径参数不同，滤波效果不一样。



(a) 原始图像



(b) 高通滤波结果



(c) 高通滤波器三维图

图 6-9 理想高通滤波消除噪声

理想高通滤波 MATLAB 代码如下：

```
%理想高通滤波程序
clc;
clear all;
I = imread('cameraman.tif');
subplot(221),
```

```
imshow(I);
title('原始图像');
s = fftshift(fft2(I));
subplot(223),
imshow(abs(s),[]);
title('图像傅里叶变换频谱');
subplot(224),
imshow(log(abs(s)),[]);
title('图像傅里叶变换取对数的频谱');
[a,b] = size(s);
a0 = round(a/2);
b0 = round(b/2);
d = 10;
p = 0.2;
q = 0.5;
for i = 1:a
   for j = 1:b
      distance = sqrt((i - a0)^2 + (j - b0)^2);
      if distance < = d h = 0;
      else h = 1;
      end;
      s(i,j) = (p + q * h) * s(i,j);
   end;
end;
s = uint8(real(ifft2(ifftshift(s))));
subplot(222),
imshow(s);
title('高通滤波结果图像');
I = imread('cameraman.tif');
[f1,f2] = freqspace(size(I),'meshgrid');
Hd = ones(size(I));
r = sqrt(f1.^2 + f2.^2);
Hd(r < 0.2) = 0;
figure
surf(Hd,'Facecolor','interp','Edgecolor','none','Facelighting','phong'); % title('理想高通滤波器');
```

6.2.2 巴特沃斯高通滤波器

巴特沃斯高通滤波器的传递函数为

$$H(u,v)=\frac{1}{1+\left[\frac{D_0}{D(u,v)}\right]^{2n}} \tag{6-10}$$

式中，D_0 为滤波器的截止频率；n 为滤波器的阶次。

巴特沃斯高通滤波器滤波性能的剖面图如图 6-10 所示。截止频率 D_0 的取值方法与巴特沃斯低通滤波器相似，该滤波其在通过频率与截止频率之间也没有明显的不连续性，图像增强后振铃效应不明显。

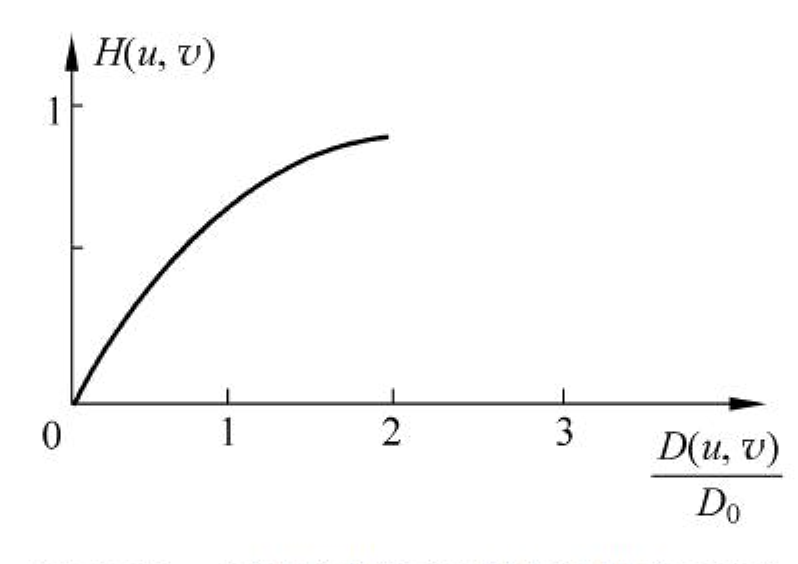


图 6-10　巴特沃斯高通滤波器剖面图

与巴特沃斯低通滤波器类似，一般情况下，取 $H(u,v)$ 下降至最大值的 1/2 时的 $D(u,v)$ 为截止频率 D_0。实际应用中，有时也取 $H(u,v)$ 下降至最大值的 0.707 时的

$D(u,v)$作为截止频率 D_0。这时其传递函数为

$$H(u,v)=\frac{1}{1+(\sqrt{2}-1)\left[\frac{D}{D_0(u,v)}\right]^{2n}} \tag{6-11}$$

6.2.3 指数高通滤波器

指数高通滤波器的传递函数为

$$H(u,v)=\mathrm{e}^{-\left[\frac{D}{D(u,v)}\right]^n} \tag{6-12}$$

其截止频率 D_0 的取值与指数低通滤波器相似,其特性曲线剖面图如图 6-11 所示。

6.2.4 梯形高通滤波器

梯形高通滤波器的传递函数为

$$H(u,v)=\begin{cases}1 & D(u,v)>D_0\\ \dfrac{D(u,v)-D_1}{D_0-D_1} & D_1\leqslant D(u,v)\leqslant D_0\\ 0 & D(u,v)<D_1\end{cases} \tag{6-13}$$

如图 6-12 所示为梯形高通滤波器滤波性能剖面示意图。

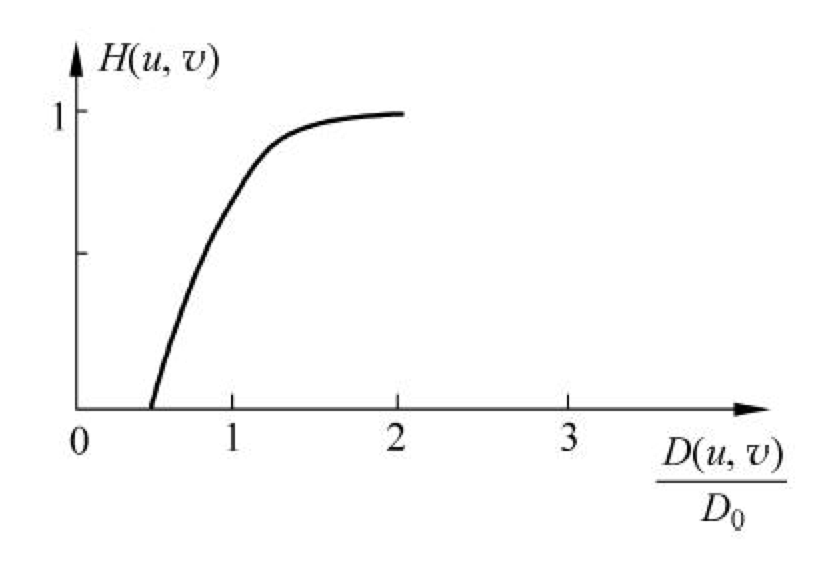


图 6-11 指数高通滤波器剖面图

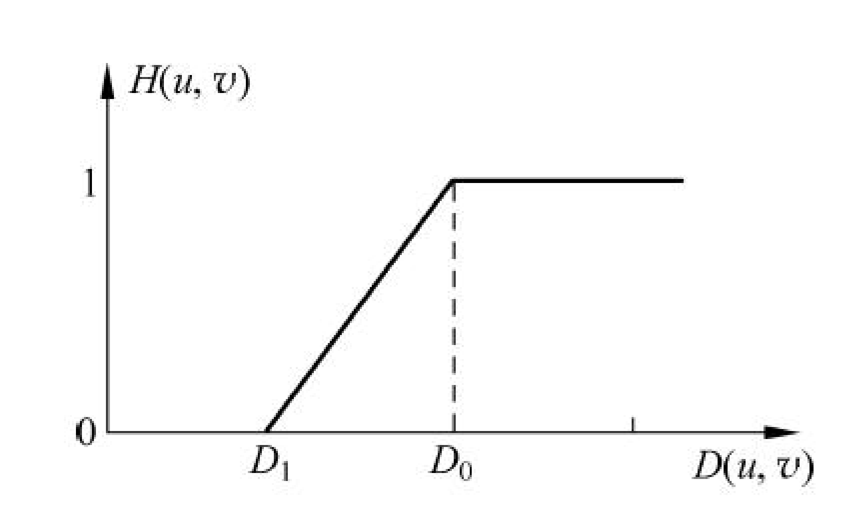


图 6-12 梯形高通滤波器剖面图

6.3 带通和带阻滤波

图像处理中,有时需要增强的信息或抑制的信息既不是图像中的高频成分也不是低频成分,而是在一个有限的频带范围内。这时,无论低通滤波器还是高通滤波器都不能完全满足使用需求,而需要采用带通或带阻滤波器。

6.3.1 带通滤波器

所谓带通滤波器,是指允许一定频率范围内的信号通过而阻止其他频率范围内的信号通过的滤波器。理想的带通滤波器的传递函数为

$$H(u,v)=\begin{cases}0 & D(u,v)<D_0-\dfrac{w}{2}\\ 1 & D_1-\dfrac{w}{2}\leqslant D(u,v)\leqslant D_0+\dfrac{w}{2}\\ 0 & D(u,v)>D_1+\dfrac{w}{2}\end{cases} \tag{6-14}$$

式中，w 为通带宽度；D_0 为通带中心频率；$D(u,v)$表示从点(u,v)到频带中心(u_0,v_0)的距离，即

$$D(u,v)=\sqrt{(u-u_0)^2+(v-v_0)^2} \tag{6-15}$$

理想带通滤波器的特性剖面图如图 6-13 所示。

6.3.2 带阻滤波器

带阻滤波器的功能是可以对一定频率范围内的信号进行完全衰减，而容许其他频率范围内的信号通过。理想的带阻滤波器的传递函数为

$$H(u,v)=\begin{cases}1 & D(u,v)<w_1\\0 & w_1\leqslant D(u,v)<w_2\\1 & D(u,v)\geqslant w_2\end{cases} \tag{6-16}$$

式中，w_1，w_2 分别为阻带开始和结束的频率，即

$$D(u,v)=\sqrt{u^2+v^2} \tag{6-17}$$

理想带阻滤波器的特性剖面图如图 6-14 所示。

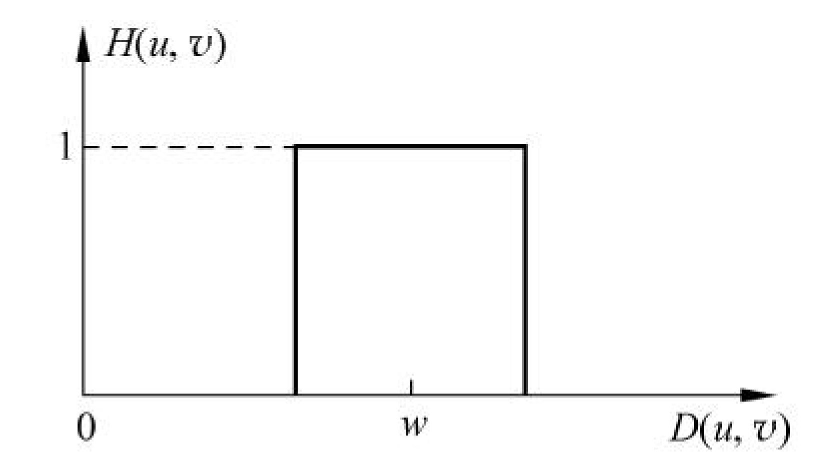


图 6-13 理想带通滤波器的特性剖面图

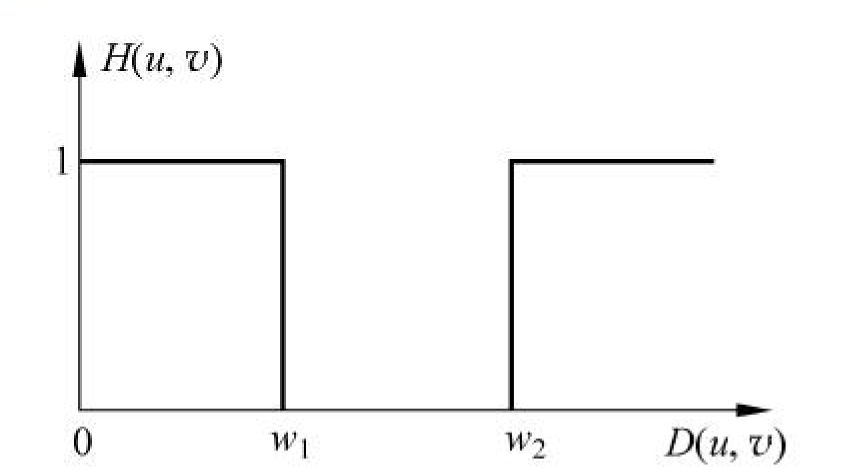


图 6-14 理想带阻滤波器的特性剖面图

6.4 同态滤波

有一类特殊的非线性系统，它遵从广义的叠加原理。在数学上，这类系统用输入和输出的矢量空间之线性变换来表示，称为同态系统。同态滤波是一种非线性滤波方法，它基于一个简单的亮度成像模型。

6.4.1 同态滤波原理

在进行图像处理及应用时，有时会遇到这样一类图像，其动态范围很大，而使用者感兴趣的图像区域的灰度值低、亮度较暗，图像细节不清晰、难以辨认，而且采用一般的灰度级线性变换处理效果不好。这时可以考虑采用非线性的同态滤波(homomorphic filtering)方法对图像进行处理。

本章前面介绍的线性滤波器对消除加性噪声很有效，但如果图像的噪声是非线性的乘性噪声，如采用发光源照明成像时，其中目标源的反射以相乘的形式对成像做出贡献，此时可以使用同态滤波。

图像的同态滤波属于频域处理范畴，其作用是对图像的亮度范围进行压缩或调整，消除图像上存在的照明不均的问题，增强图像暗区的细节信息，同时又不损失亮区的图像细节，从而实现对图像的对比度进行增强。

简而言之，同态滤波是利用广义叠加原理对系统进行滤波，是一种频域滤波和灰度变换结合的图像处理方法，它以图像的照度/反射率模型作为频域处理的基础，通过压缩图像亮度范围和增强对比度来改善图像质量。使用这种方法处理图像非常符合人眼对于亮度响应的非线性特性，避免了直接对图像进行傅里叶变换时产生的失真。

同态滤波的基本原理：将像元灰度值看作照度和反射率两个组分的产物。由于照度相对变化很小，可以看作图像的低频成分，反射率则是高频成分。通过分别处理照度和反射率对像元灰度值的影响，实现增强暗区细节特征的目的。

6.4.2 同态滤波实现方法

应用同态滤波消除图像噪声，应先利用对数变换将乘性噪声转化为加性噪声，通过线性滤波器消除噪声后再进行非线性的指数反变换得到无噪声的原始图像。同态滤波处理的基本流程如图 6-15 所示。

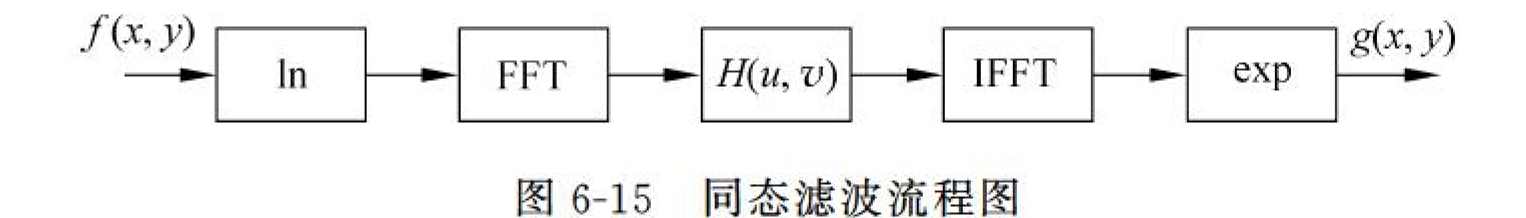


图 6-15 同态滤波流程图

图 6-15 中，$f(x,y)$表示原始图像；$g(x,y)$表示处理后的图像；ln 代表对数运算；FFT 代表傅里叶变换；$H(u,v)$表示同态滤波；IFFT 代表傅里叶逆变换；exp 代表指数运算。具体过程如下：

(1) 对 $f(x,y)$进行对数变换。

一般自然景物的图像 $f(x,y)$可由照明(illumination)函数 $i(x,y)$和反射(reflectance)函数 $r(x,y)$的乘积表示。$i(x,y)$描述景物的照明，与景物无关；$r(x,y)$包含景物的细节，与照明无关。

即

$$f(x,y)=i(x,y)r(x,y)$$
$$0<i(x,y)<\infty;0<r(x,y)<1 \tag{6-18}$$

由于 i 与 r 函数二者相乘，先变换到频域再对其进行处理，因此对上式取对数得

$$\ln f(x,y)=\ln i(x,y)+\ln r(x,y)$$

(2) 对上式进行傅里叶变换。

$$F(u,v)=I(u,v)+R(u,v) \tag{6-19}$$

$I(u,v)$照明函数在空间上变化缓慢，其频谱特性集中在低频段；$R(u,v)$反射函数的频谱集中在高频段，景物本身具有较多的细节和边缘，反射函数描述的景物，反映图像的细节内容，其频率处于高频区域。

(3) 采用频域增强函数 $H(u,v)$处理 $F(u,v)$。

$$H(u,v)F(u,v)=H(u,v)I(u,v)+H(u,v)R(u,v) \tag{6-20}$$

若图像照明不均，则图像内各部分的平均亮度会有起伏，这时图像暗区的细节较难分

辨，因此，需要消除这种不均匀性。同态滤波方法是压缩照明函数的灰度范围，即在频域削弱照明函数的成分，同时增强反射函数的频谱成分，因此增加了反映图像对比度的反射函数的比重。因而使图像暗区的图像细节信息得以增强，而且还尽可能地保持了亮区的图像细节。通过与传递函数同态滤波器 $H(u,v)$ 相乘，低频段被压缩，而高频段被扩展。

（4）进行 IFFT 变换。

进行傅里叶反变换得到对应空间域的表达式为

$$h_f(x,y) = h_i(x,y) + h_r(x,y) \tag{6-21}$$

根据上式可以看出，增强以后的图像分别对应照度分量和反射分量两部分，图像由两部分叠加而成。

（5）对上式取指数函数。

$$g(x,y) = \exp[h_f(x,y)] = \exp[h_i(x,y) + h_r(x,y)] \tag{6-22}$$

根据不同的图像特性和需要，选用不同的 $H(u,v)$，可得到比较理想的滤波结果。对于细节对比度差、分辨不清的图像采用同态滤波器处理后，图像的亮度比较均匀，细节得到一定的增强。

6.4.3　同态滤波的应用

例 6-4：数字图像同态滤波效果实验

根据同态滤波算法的原理，其 MATLAB 程序代码如下：

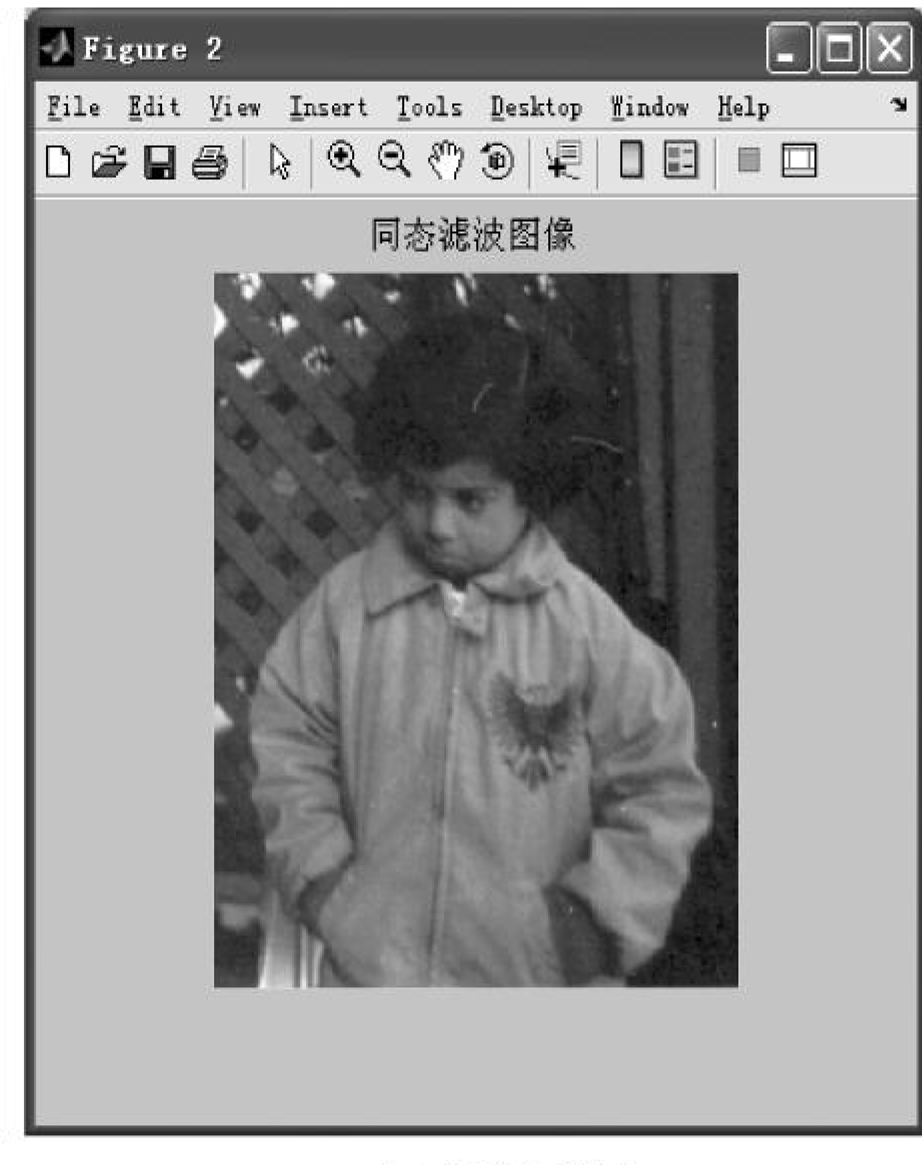
```
clear;
close all;
[name,path] = uigetfile('pout.bmp');
file = strcat(path,name);
[X,map] = imread(file);
X = double(X);
I = rgb2hsv(X);
H = I(:,:,1);
S = I(:,:,2);
V = I(:,:,3);
figure;
imshow('pout.bmp');
title('原始图像');
%构造一个高斯滤波器
f_high = 1.0;
f_low = 0.8;                                    %得到一个高斯低通滤波器
gauss_low_filter = fspecial('gaussian', [7 7], 1.414);
matsize = size(gauss_low_filter);
%同态滤波滤除高频部分, 将低通滤波器转换为高通滤波器
%f_high 和 f_low 为高通滤波器形态参数
gauss_high_filter = zeros(matsize);
gauss_high_filter(ceil(matsize(1,1)/2) ,ceil(matsize(1,2)/2)) = 1.0;
gauss_high_filter = f_high * gauss_high_filter - (f_high - f_low) * gauss_low_filter;
%对数变换将入射光和反射光分开
log_img = log(double(V) + eps);
%将高斯高通滤波器与对数转换后的图像卷积
high_log_part = imfilter(log_img, gauss_high_filter, 'symmetric', 'conv');
```

```
%指数 exp 变换
high_part = exp(high_log_part);
minv = min(min(high_part));
maxv = max(max(high_part));
%结果图像
temp = (high_part - minv)/(maxv - minv);
S = adapthisteq(S) * 2.1;
HSV3(:,:,1) = H;
HSV3(:,:,2) = S;
HSV3(:,:,3) = temp;
rgb2 = hsv2rgb(HSV3);                                    %转换到 RGB 空间
figure;
imshow(rgb2);
title('同态滤波结果');
```

同态滤波程序运行结果如图 6-16 所示。



(a) 原始图像



(b) 同态滤波结果

图 6-16 同态滤波图像处理

习题

1. 图像频域增强的原理是什么?

2. 试阐述低通滤波的原理和作用。

3. 根据空域滤波增强和频域滤波增强的基本理论,分析空域滤波中的邻域均值滤波和低通滤波方法的特点。

4. 试比较二维低通巴特沃斯滤波器和二维低通梯形滤波器的特点有何异同。

5. 简述高通滤波的原理及特点。

6. 挑选一幅图像,编写 MATLAB 频域滤波程序,实现如下功能:

(1) 对图像进行低通滤波。

(2) 对图像进行高通滤波。

(3) 观察低通、高通滤波的实验结果,分析两种滤波方法的功能和应用范围。

7. 简要分析频域带通滤波的应用条件。

8. 选择一幅图像,编写 MATLAB 程序对其进行带通和带阻滤波,并分析滤波效果。

9. 试阐述同态滤波的原理和实现过程。

10. 选择一幅图像进行同态滤波,分析同态滤波的作用与效果。

第7章 彩色图像处理

CHAPTER 7

随着信息技术和人工智能技术的发展，在图像信息中，彩色图像占据的比例越来越高，其应用已越来越广泛。人类视觉对灰度级数的辨别能力有限，而对彩色的识别能力强，因此，与灰度图像相比，彩色图像携带了更多的特征和可视化信息，彩色图像处理已成为图像处理学科的一个重要的研究分支。

本章首先介绍人眼视觉系统及其特点，然后介绍计算机系统的颜色模型，包括面向硬件设备的颜色模型(RGB和CMYK)和面向视觉特性的颜色模型(HSI、YUV、HSV和Lab)，最后介绍彩色图像处理的基本原理和方法。

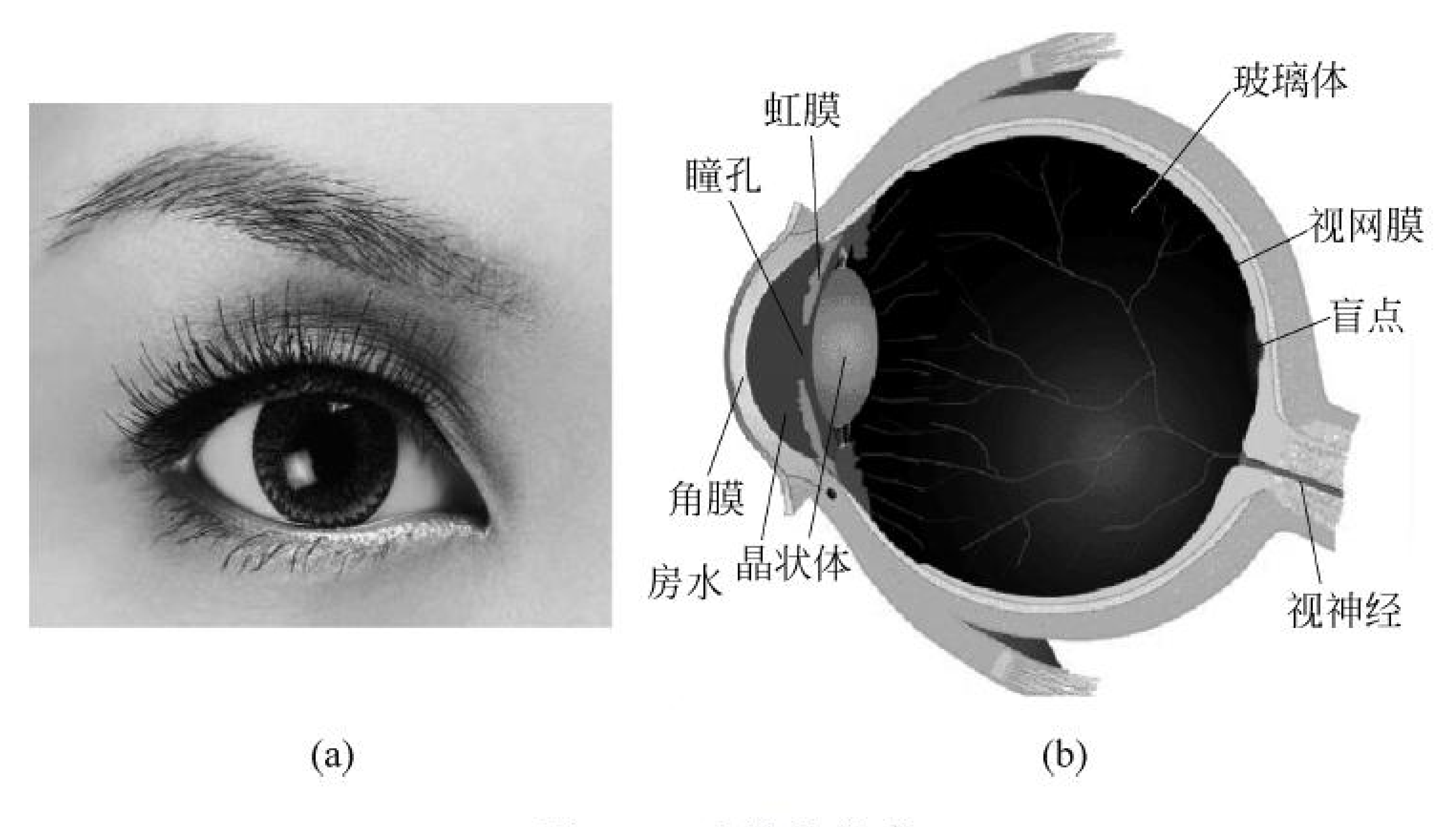
7.1 人眼的视觉特性

人类在长期的进化过程中，形成了特有的视觉特性。人类视觉系统是世界上最好的图像处理系统，即便如此，人眼也不是完美的。人眼视觉系统对图像的认知是非均匀、非线性的，并不是对图像中的任何变化都能感知。例如，在一定范围内，人眼不能觉察到图像系数的量化误差所产生的图像差异。因此，若图像编码方法能利用人类视觉系统的这些特点，则可以得到更高压缩比的方法。对人眼视觉特性的深入研究以及基于视觉特性而建立的各种数学模型，一直是数字图像各种压缩算法的基础。

7.1.1 人眼的结构

人的眼睛主要由视网膜、瞳孔、角膜、晶状体等组成，如图7-1所示。其中视网膜是人眼感受光的部分，视网膜就像一架照相机的感光底片，专门负责感光成像。边缘部位杆状细胞占多数，中央部位锥状细胞占多数。这两种细胞对光的感受特性分工不同，杆状细胞对光的感受性比锥状细胞要高。因此，若视觉亮度大约在$10^{-6}\sim10^{-2}\text{cd/m}^2$的照度范围，只有杆状细胞在工作，这种状态被称为暗视觉。当亮度达到10cd/m^2时锥状细胞开始工作，当亮度在$10^{-2}\sim10\text{cd/m}^2$时，杆状细胞和锥状细胞同时起作用，这种视觉状态被称为中介视觉。杆状细胞感受光，锥状细胞感受颜色，因此，只有在照度较高的条件下，人的视觉才会有良好的颜色感。

由于杆状细胞和锥状细胞是在神经系统的综合控制下而起作用，使得人眼具有视觉适应等一系列独有的特性。视觉适应是视觉器官的感觉随外界亮度的刺激而变化的过程，即



(a) (b)

图 7-1 人眼的组成

现在和过去呈现的各种亮度、光谱分布、视角的刺激下，视觉系统状态的变化，有时也指这一过程达到的最终状态。视觉适应的机制包括视细胞或神经活动的重新调整、瞳孔的变化及明视觉与暗视觉功能的转换。

视觉适应分为明适应和暗适应，暗适应所需过渡时间较长，明适应过程较短，一般为2～3min。视觉不会瞬时产生，也不会瞬时消失，尤其在遇到高亮度的闪光后，人眼往往会感受到一连串影像，以不规则的强度和不断降低的频率正负交替出现，这种现象被称为后像。人的眼睛突然遇到强光照射时，在之后一段时间内，会被一个黑影困扰。因此，强光对人眼视觉具有一定的危害。使人眼感到不舒服的还有眩光，眩光也可分为两类，一是失能眩光，二是不舒适眩光。不舒适眩光会使人产生烦躁的感觉，失能眩光会导致眼部疲劳，工作效率降低，甚至可致短暂致盲。光源的发光颜色与温度有关，如白炽灯，灯丝温度低，发出红光，温度高，发出白光。当光源照射到物体上时，该物体色彩与阳光照射时的色彩不完全一致，有一定的失真度。

人眼视觉特性表明，良好的照明质量，对提高学习和工作效率及减少视觉疲劳具有很重要的作用。适合人类视觉的照明光源，首先是照度的均匀性，不均的照度容易引起视力疲劳，这就要求室内的最大和最小照度与平均照度之差的绝对值不大于平均照度的 1/6。其次，要考虑照度的稳定性，若照度波动大，特别是每秒 20 次下的严重波动，对眼睛危害很大，应避免。

7.1.2 人眼视觉模型

人类视觉系统的工作原理类似于光学系统，但它不是普通意义上的光学系统，还受到神经系统的调节，因此远比光学系统复杂。为了对人类视觉系统进行研究与分析，可以用光学系统的一些原理来解释人眼的一些视觉特性。

目前认为，人眼视觉模型是一个低通-对数-高通模型，如图 7-2 所示。

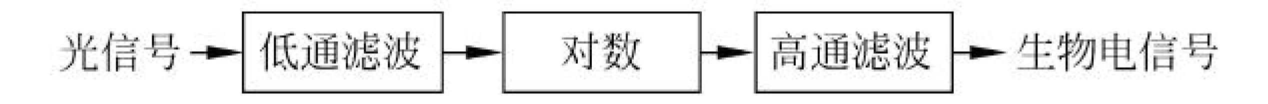


图 7-2 低通-对数-高通人眼视觉模型

该模型分为三个阶段：

（1）第一阶段，是人眼接收光信号的过程。在这一阶段，由于人眼晶状体光学偏差，视觉系统的上限频率起到作用，人眼对高频变化不敏感，相当于对信号进行低通滤波。

（2）第二阶段，是视觉细胞响应光信号的过程。由于人眼视觉系统的生物结构，视觉的感受亮度和空间的客观亮度关系不能近似为数学上的线性函数关系，而与非线性的对数函数关系非常接近。

（3）第三阶段，是视神经细胞的解析过程阶段。由于视神经细胞的侧向抑制作用，人眼对低频光不敏感，该过程等效于高通滤波。

根据上述模型，人眼观察图像时表现出以下几个方面的特性：

（1）从空间频率域看，人眼是一个线性低通滤波系统，分辨景物的能力是有限的。由于瞳孔有一定的几何尺寸和一定的光学像差，视觉细胞有一定的大小，所以人眼的分辨率不能是无穷的，HVS（人眼视觉系统）对太高的频率不敏感。

（2）人眼对亮度的响应具有非线性对数特性，以提高其亮度的动态范围。由于人眼对亮度响应的这种非线性特性，在平均亮度大的区域，人眼对灰度小幅变化不敏感。

（3）由于人眼受神经系统的调节，从空间频率的角度来说，人眼视觉系统不仅是一个线性系统，而且是一个具有带通特性的系统。由信号分析的理论，人眼视觉系统对信号进行加权求和运算，相当于使信号通过一个带通滤波器，结果会使人眼产生一种边缘增强感觉——侧抑制效应。人眼对亮度信号的空间分辨率大于对色度信号的空间分辨率。

（4）图像的边缘信息对视觉很重要，特别是边缘的位置信息。人眼容易感觉到边缘的位置变化，而对于边缘的灰度误差，人眼并不敏感。

由于人眼系统的复杂性，实际上，人眼的视觉特性也是一个多信道（multichannel）模型，或者说，具有多频信道分解特性（mutifrequency channel decompositon）。

7.1.3 人类视觉特性

在宽泛的电磁波谱中，人眼仅仅可感知其中很窄范围的一段波长，这一段电磁波谱称为可见光，即可见光是指电磁波谱中人眼可以感知的部分。人眼可以感知的波长范围如图 7-3所示。由于人眼结构的复杂性和个体差异，可见光谱没有精确的波长范围，一般认为波长在 380～780nm 范围的光为可见光。

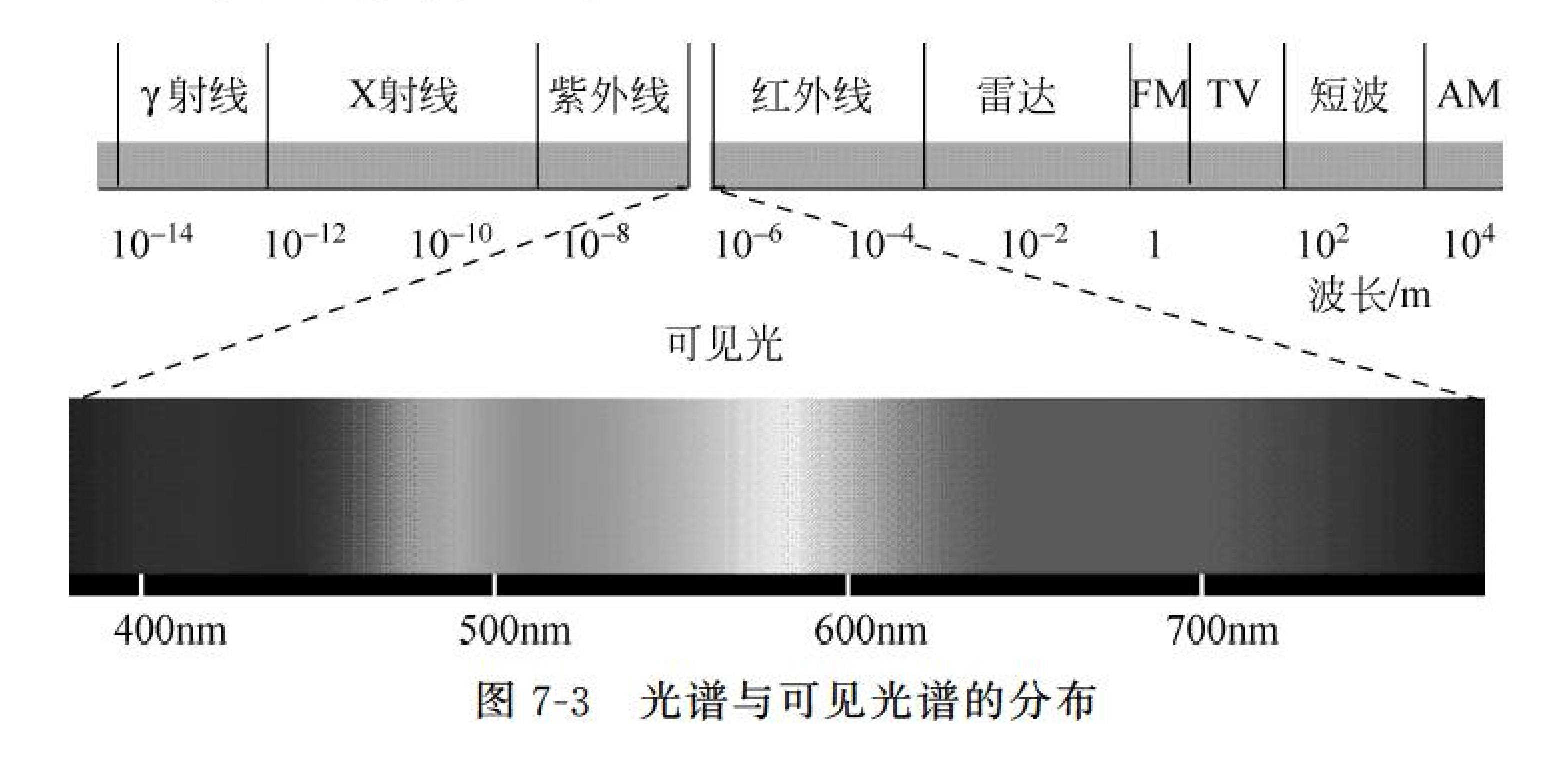


图 7-3 光谱与可见光谱的分布

即便对于 380～780nm 波长范围的可见光，人眼对不同波长的光具有不同的敏感程度，这一特性称为人眼的视敏特性。描述人眼视敏特性的物理量有视敏函数和相对视敏函数。

（1）视敏函数。在相同亮度感觉的条件下，不同波长的光辐射功率的倒数称为视敏函数，即

$$k(\lambda) = \frac{1}{P_r(\lambda)} \tag{7-1}$$

（2）相对视敏函数。相对视敏函数是基于比值定义的，大量的实验研究表明，人眼对波长为555nm的光最敏感，任一波长的光的视敏函数与最大视敏函数值 $K(555)$ 的比值称为相对视敏函数，即

$$V(\lambda) = \frac{K(\lambda)}{K_{\max}} = \frac{K(\lambda)}{K(555)} = \frac{P_r(555)}{P_r(\lambda)} \tag{7-2}$$

研究表明，视敏函数曲线的最大值位于555nm处，当光线微弱向左偏移至最大值为507nm处，HVS就相当于带通滤波器，这就表明人眼对亮度变化比较敏感。

人眼对于蓝光的视觉灵敏度要比红光和绿光低得多，人眼三种锥状细胞相对视敏函数曲线如图7-4所示。三条曲线的峰值比为 $R:G:B=0.54:0.575:0.053$（蓝光放大20倍）。三条曲线有相当一部分是重叠的。正常观察条件下，人眼得到的是合成视觉特性，不能感受到三者各自的数值，根据三者的比例，HVS感知彩色的色调和饱和度，由三者之和确定光的总亮度和颜色。

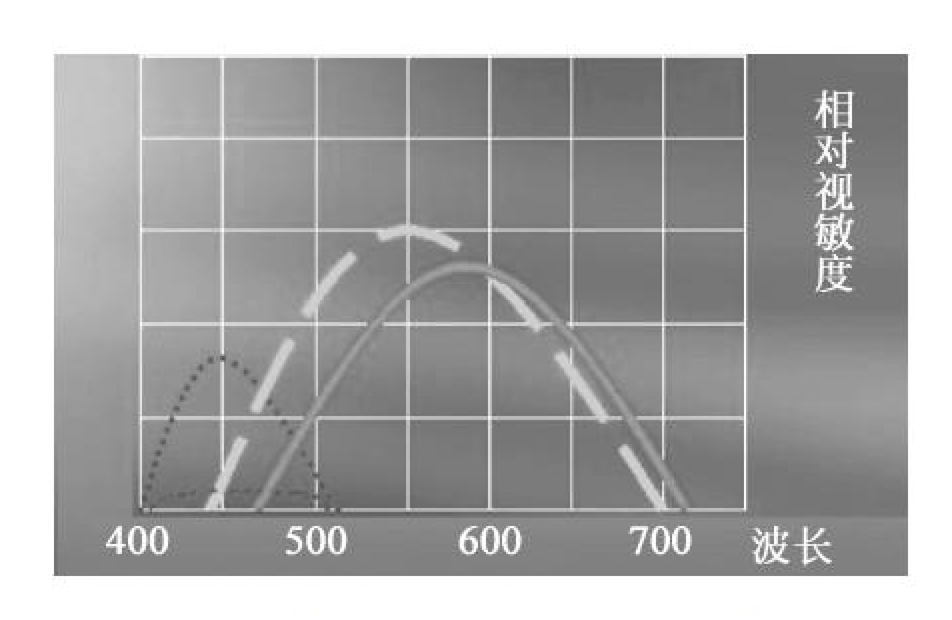


图7-4　三种锥状细胞相对视敏函数曲线

长期以来，通过对人眼视觉现象的实验和研究，发现人眼的视觉特性具有对比灵敏度、分辨率、马赫带效应、视觉掩盖效应等很多特点，这些特性可以直接或间接地用于改善图像与视频信息的处理。如何充分利用人眼的视觉特性，已成为现代编码技术中首先要考虑的一个基本问题。

1）对比灵敏度

在可见光范围内，人眼对光强变化的响应是非线性的，通常将人眼主观感受上可辨别的亮度差所需的最小光强差值称为亮度的可见度阈值。也就是说，当光强 I 从某一值开始逐渐增大，在一定幅度内人眼感觉不到亮度的变化，必须变化到一定增量 ΔI 时，人眼才能感觉到亮度发生了变化，$\Delta I/I$ 称为对比灵敏度。因此如果恢复图像的误差低于对比灵敏度，人眼就感觉不到图像质量有变化。人眼对于运动图像的对比灵敏度与时间轴上信息的变化速度有关，随着时间轴变化频率的增加，人眼所能感受到的图像信息的误差阈值呈上升趋势，视觉上的这种动态对比灵敏度特性表现为图像序列之间相互掩盖效应。在图像编码上，可见度阈值和掩盖效应对量化器的设计具有重要意义，利用这一视觉特性，在图像的边缘可以容忍较大的量化误差，使量化级数减少，从而提高图像的编码效率。

2）分辨率

当空间平面上两个黑点不断靠近到某一距离时，距黑点一定距离的观察者将无法区分，这说明人眼分辨景物细节的能力是有限的，可区分的最小距离就是分辨率。研究表明，人类视觉的分辨率具有如下特点：当照度太强、太弱，或背景亮度太强时，分辨率降低；当目标快速运动时，分辨率降低。对彩色细节的分辨率比对亮度细节的分辨率低，如果黑白分辨率为1，则黑红为0.4，绿蓝为0.19。

3）马赫带效应

马赫带效应(Mach band effect)是1868年由奥地利物理学家E.马赫发现的一种明度对比现象，它是一种主观的边缘对比效应。所谓马赫带现象，是指人眼观察两块亮度不同的区域时，边界处的亮度对比会感觉到更大，使轮廓表现得特别明显，如图7-5所示。



图7-5 马赫带效应示意图

马赫带效应的产生是人类的视觉系统特性造成的，HVS主观感受在亮度变化之处出现虚幻的明亮或黑暗的条纹。生理学对马赫带效应的解释是：人类的视觉系统有增强边缘对比度的特性。当亮度发生跃变时，会产生边缘增强的感觉，视觉上会感到亮侧更亮，暗侧更暗。人类视觉的马赫带效应会导致局部阈值效应，即在边缘的亮侧，靠近边缘像素的误差感知阈值比远离边缘阈值高3～4倍，边缘掩盖了其邻近像素，因此对靠近边缘的像素编码误差可以适度大一些。

7.2 计算机的颜色模型

自然界是一个具有丰富色彩的世界，不仅不同的物体有不同的颜色，而且相同的颜色还具有不同的色度和饱和度。计算机系统的彩色模型又称为彩色空间或彩色坐标系。图像处理中的颜色模型有很多种，常用的颜色模型有RGB颜色模型、HSI颜色模型、HSV颜色模型、CMYK颜色模型、YUV颜色模型、Lab颜色模型。

7.2.1 面向硬件的彩色模型

面向硬件的颜色模型主要包括RGB和CMYK颜色模型。由于这类颜色模型面向硬件设备，非常适合图像采集输入和图像输出显示的场合使用。

1. RGB模型

自20世纪60年代以来，人类对计算机的彩色表示进行了系统深入的研究，科学研究表明，任何色彩都可以用R、G、B三种颜色按一定的比例混合叠加实现。实际上，人眼的锥状感光细胞将可见光部分分为R、G、B三个波段，因此R、G、B又称为三基色。如图7-6所示为视觉系统三类锥状感光细胞的光谱敏感曲线。

R、G、B称为人类视觉的三基色，RGB彩色系统在日常生活的应用非常广泛。例如，人们常用的彩色打印机就是通过红、绿、蓝三色叠加在一起打印，从而显示出不同的颜色；彩色照相时，通过三层不同的摄影乳剂合成获得真彩色图像；在计算机中，每个像素的颜色同样是由RGB三原色(即三个分量)按不同比例叠加实现的，每一分量的亮度值的大小限定在一定范围，如[0,255]。RGB(0,0,0)表示黑色，RGB(255,255,0)表示由红绿叠加得到黄

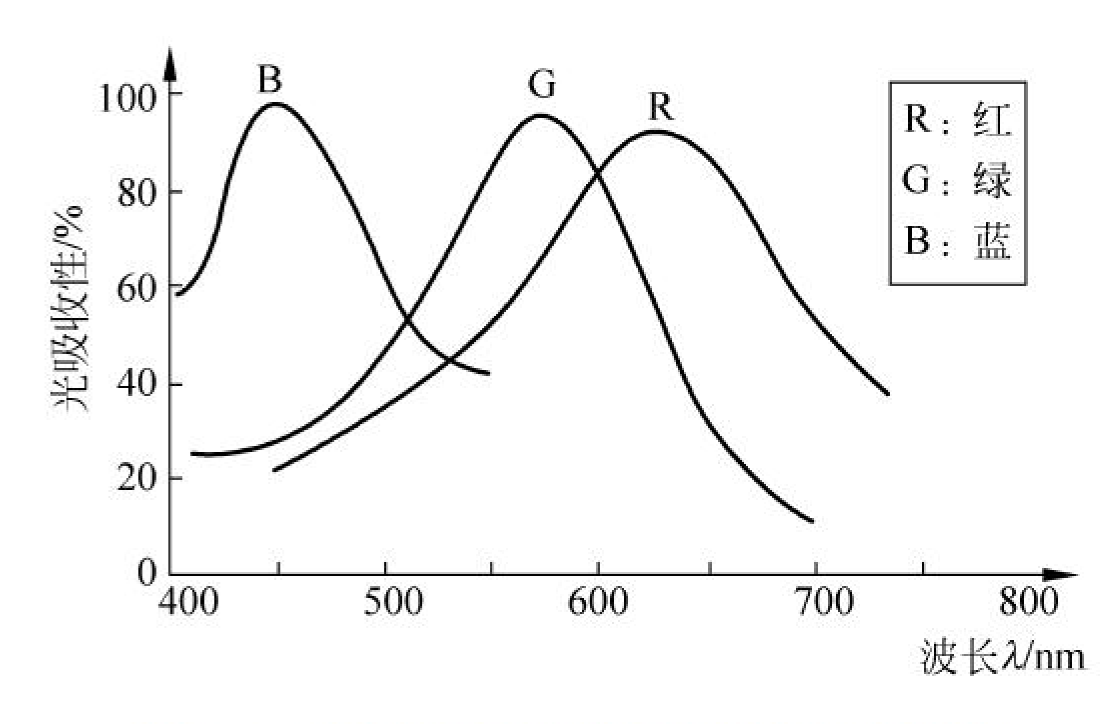


图 7-6 锥状感光细胞的电磁波谱响应曲线

色，而 RGB(255,255,255)表示由红绿蓝叠加得到白色。在 RGB 彩色空间的原点上，三个基色均没有亮度，即原点为黑色，三基色达到最高亮度时表现为白色，如图 7-7 所示。亮度较低的等量的三种基色产生灰色的基调，所有这些点均落在彩色立方体的对角线上，该对角线被称为灰色线。

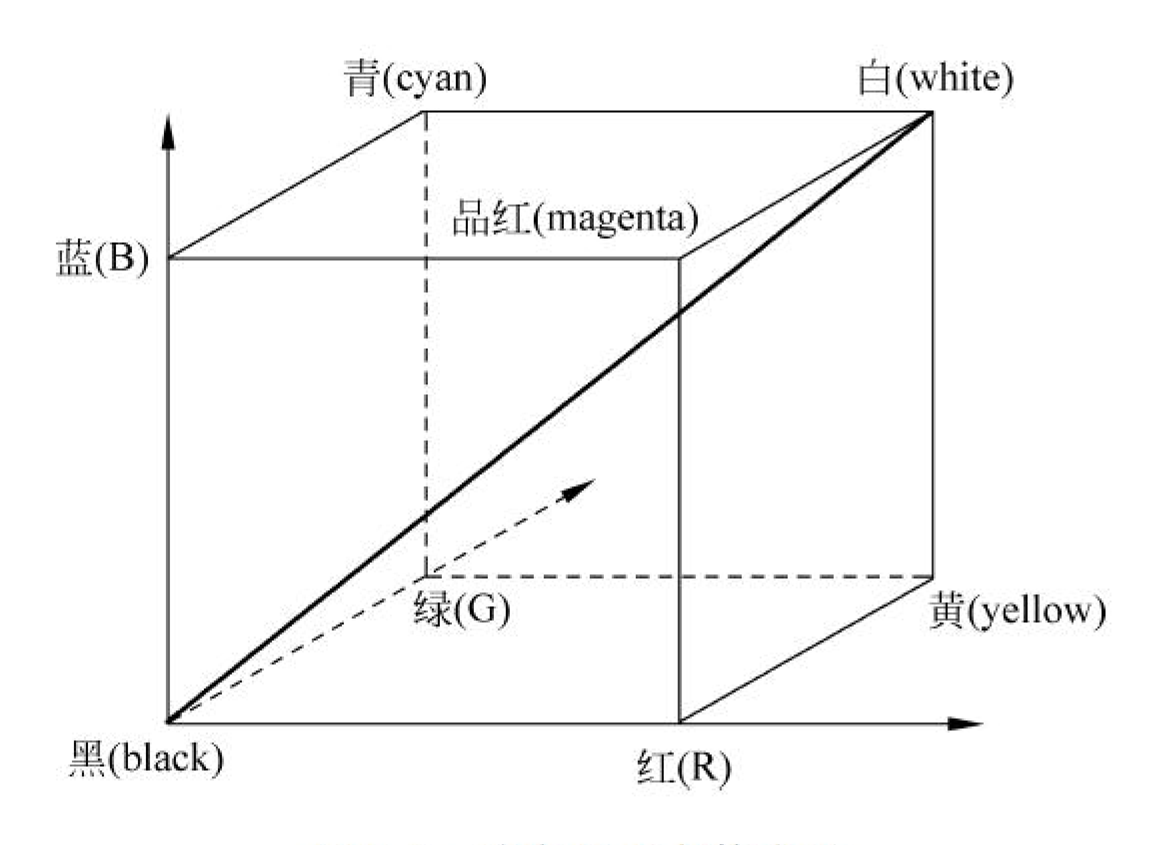


图 7-7 彩色的立方体表示

2. CMYK 颜色模型

CMYK 表色系统也是一种常用的颜色表示模型。计算机屏幕显示通常用 RGB 系统，它是通过三基色叠加来产生丰富多彩的颜色，这种做法通常称为加色合成法。而在印刷工业上则通常用 CMYK 表色系统，它是通过颜色相减来产生其他颜色的，所以称这种方式为减色合成法。

CMYK 模式的原色为青色(cyan)、品红色(magenta)、黄色(yellow)和黑色(black)。由于 CMYK 模式的彩色图像文件比其他彩色表示方法的文件要大，占用了太多的计算机磁盘空间和内存，因此数字图像处理中一般不用 CMYK 模式，该模式一般在彩色印刷领域使用。在 CMYK 模型中，如去除黑色，就是 CMY 颜色模型，同样是基于减色法合成彩色原理。

7.2.2 面向视觉感知的颜色模型

面向硬件设备的颜色模型与人的视觉感知有一定的差距，使用时不太方便，最典型的问题是，给定一个彩色图像，人眼难以确定其 RGB 分量，这时面向视觉感知的彩色模型比较方

便。这些模型独立于显示设备，与人类颜色视觉感知比较接近。

1. HSI 颜色模型

与 RGB 格式从物理和光学角度描述颜色不同，HSI 则是根据视觉的主观感觉对颜色进行描述。研究和实践均表明，人眼不能直接感觉红、绿、蓝三色的比例，只能通过感知颜色的亮度、色调和饱和度来区分物体。在 HSI 彩色空间中，I 表示强度(intensity)，它决定了像素点的整体亮度，而不考虑色彩，而表征像素彩色信息的两个参数是色度(hue)和饱和度(saturation)。

色度 H 表明颜色的种类，而颜色是由波长决定的，所以彩色的色度 S 表示该彩色最接近什么样波长的光谱，不同的波长显示出不同的颜色，如波长为 0.55μm 的呈现绿色，植物由于叶片中含有叶绿素，在 0.55μm 左右有一个反射高峰，使叶子看上去呈现绿色。据研究，人眼所能分辨的自然界的色度大约有 120 种。

饱和度是表示颜色深、浅、浓、淡程度的物理量。例如，同是红色，却有深红和浅红之分。HSI 颜色模型反映了人的视觉对色彩的感觉，Castleman 博士在《数字图像处理》中用一桶颜料作了形象的注释：假设有一桶纯红色的颜料，其饱和度为 1，如果掺入白色染料后，则红色变淡，即减少了它的饱和度，随着更多的白色染料加入，饱和度逐渐降低，直至变成白色，即饱和度为 0。图像处理中，通常用一个柱体坐标系来表示 HSI 空间，如图 7-8 所示。

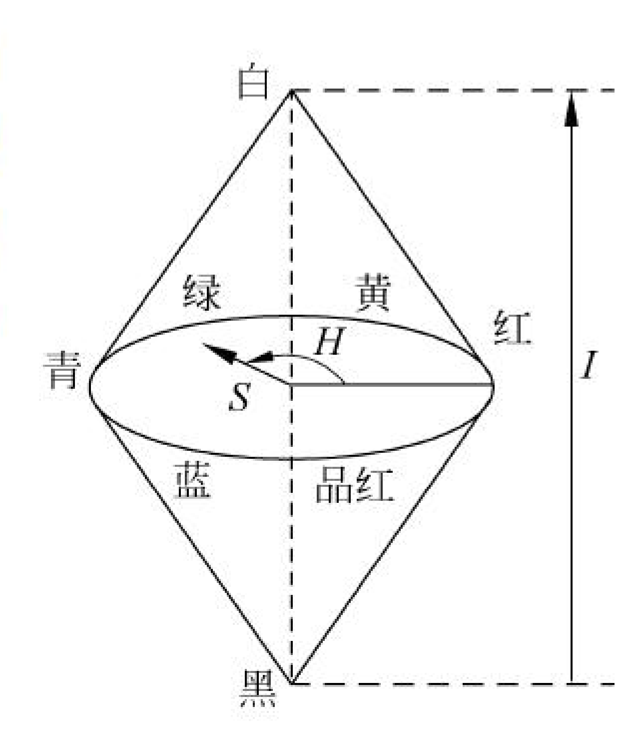


图 7-8　HSI 彩色空间坐标系

柱体高度表示强度分量 I，柱体的圆环表示色度分量 H，以“度”为衡量单位，柱体的半径表示饱和度分量 S，半径越大，S 越大。具有最高亮度和最大饱和度的颜色位于圆柱上底圆的圆周上。不失一般性，计算机中，强度(I)的值域为[0,1]，色度(H)的值域为[0,360]或[0,1)，饱和度(S)的值域则为[0,1]。

I 是人眼所能感觉到的表示颜色明暗程度的物理量，即强度量。在数字图像处理中，为了便于两种色彩空间之间的变换，一般将 I 定义为 RGB 三个分量的平均值(有的变换公式取 RGB 的最大值)，因此 I 确定了像素的整体亮度。

RGB 与 HSI 模型之间是可以互相转换的，对于归一化到[0～1]的 R、G、B 颜色体系，由 RGB 到 HSI 的转换公式为

$$\begin{cases} I = \dfrac{1}{3}(R+G+B) \\ S = I - \dfrac{3}{R+G+B}[\min(R,G,B)] \\ H = \begin{cases} \arccos\left\{\dfrac{[(R-G)+(R-B)]/2}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\} & R \neq G \text{或} R \neq B \\ 360^\circ - \arccos\left\{\dfrac{[(R-G)+(R-B)]/2}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\} & B > G \end{cases} \end{cases} \tag{7-3}$$

由 HSI 到 RGB 的转换公式分为 3 种情况：

(1) 若 H 在 0°～120°之间,转换公式为

$$\begin{cases} R = I\left[1 + \dfrac{S\cos(H)}{\cos(60° - H)}\right] \\ G = 3I - (B + R) \\ B = I(1 - S) \end{cases} \tag{7-4}$$

(2) 若 H 在 120°～240°之间,转换公式为

$$\begin{cases} R = I(1 - S) \\ G = I\left[1 + \dfrac{S\cos(H - 120°)}{\cos(180° - H)}\right] \\ B = 3I - (R + G) \end{cases} \tag{7-5}$$

(3) 若 H 在 240°～360°之间,转换公式为

$$\begin{cases} R = 3I - (G + B) \\ G = I(1 - S) \\ B = \left[1 + \dfrac{S\cos(H - 240°)}{\cos(300° - H)}\right] \end{cases} \tag{7-6}$$

2. HSV 颜色模型

HSV(hue, saturation, value)颜色模型是 A. R. Smith 根据颜色的直观特性于 1978 年定义的一种颜色空间,可以用到圆锥体表示,也可以用六角锥体模型表示(也可称为六角锥体模型)。不同于 RGB 和 CMYK 面向硬件的特性,HSV 颜色模型是面向视觉感知的。RGB 颜色空间中,三种颜色分量的取值与所生成的颜色之间的联系并不直观,而 HSV 颜色空间,更类似于人类感觉色彩的方式。

HSV 模型包括色度(hue)、饱和度(saturation)、明度(value 或 brightness)三个参数,是一种将 RGB 色彩空间中的点在倒圆锥体中的表示方法。

色度 H 是色彩的基本属性,即所处的光谱颜色的位置,就是平常所指的颜色,如红色、黄色等,H 取值范围为 0°～360°,从红色开始按逆时针方向的顺序,红色为 0°,绿色为 120°,蓝色为 240°,它们的补色分别是黄色为 60°,青色为 180°,品红为 300°。

饱和度 S 为一比例值,也可以称为纯度,表示该颜色接近光谱色的程度,即色彩的纯度,取值 0～100%之间,值越高色彩越纯,值低则逐渐变灰。饱和度高,颜色则深而艳。

明度 V 表示颜色明亮的程度,通常取值范围为 0～100%,即由黑至白。它和光强度之间并没有直接的联系,对于光源色,明度值与发光体的光亮度有关;对于物体色,此值和物体的透射比或反射比有关。

HSV 颜色空间可以用一个圆锥空间模型来描述,如图 7-9 所示,圆锥的顶点处,$V=0$,H 和 S 无定义,代表黑色。圆锥的顶面中心处 $V=\max$,$S=0$,H 无定义,代表白色。HSV 的三维表示可以认为是从 RGB 立方体演化而来,设想从 RGB 沿立方体对角线的白色顶点向黑色顶点观察,就可以看到立方体的六边形外形。六边形边界表示色彩,水平轴表示纯度,明度沿垂直轴测量。

RGB 与 HSV 模型之间也可以互相转换。

(1) 由 RGB 到 HSV 的转换公式为

令

$$\max = \max(R, G, B)$$

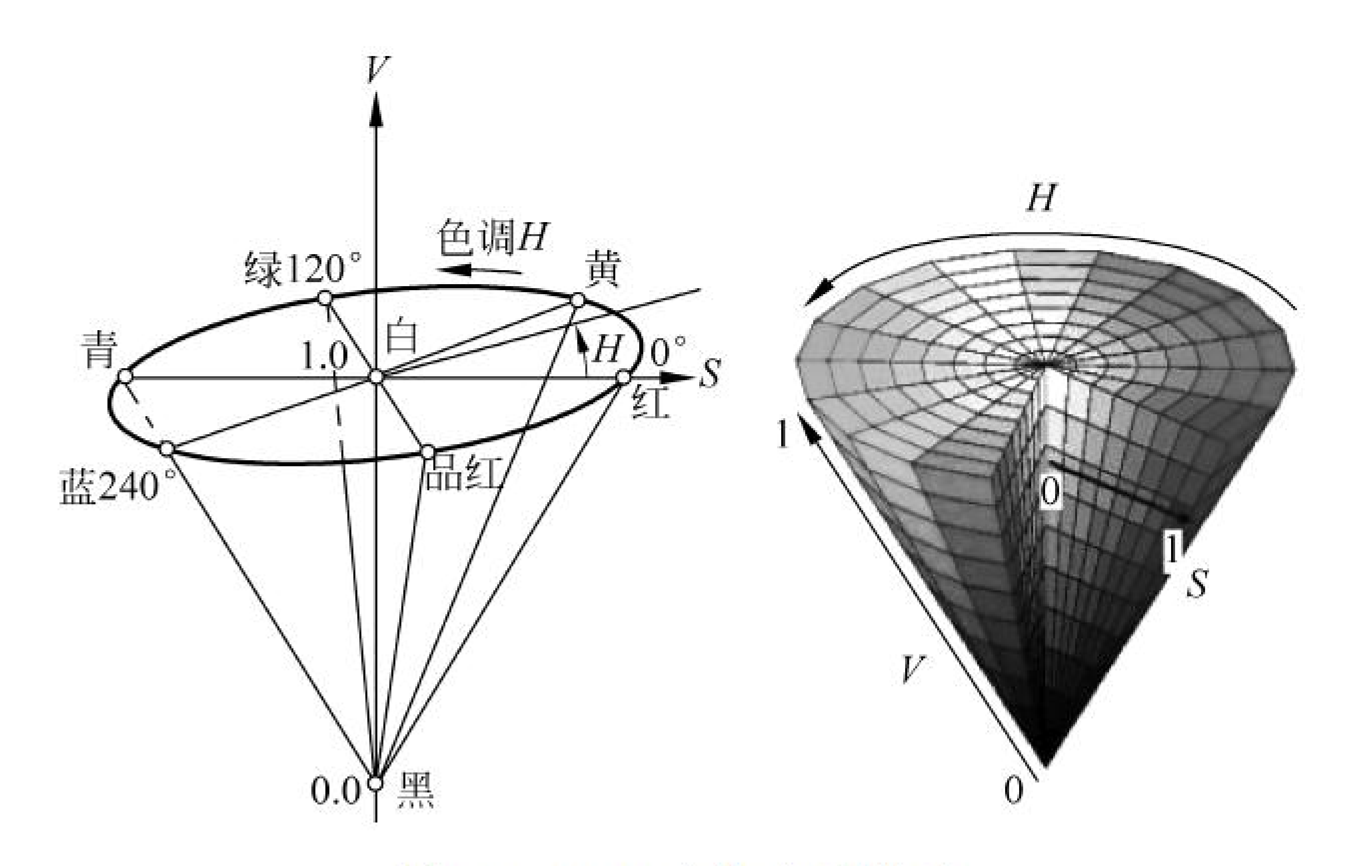


图 7-9 HSV 颜色空间模型

$$\min = \min(R, G, B)$$

$$h = \begin{cases} 0^\circ & \max = \min \\ 60^\circ \times \dfrac{g-b}{\max - \min} + 0^\circ & \max = r, g \geqslant b \\ 60^\circ \times \dfrac{g-b}{\max - \min} + 360^\circ & \max = r, g < b \\ 60^\circ \times \dfrac{b-r}{\max - \min} + 120^\circ & \max = g \\ 60^\circ \times \dfrac{r-g}{\max - \min} + 240^\circ & \max = b \end{cases} \tag{7-7}$$

$$s = \begin{cases} 0^\circ & \max = 0 \\ \dfrac{\max - \min}{\max} = 1 - \dfrac{\min}{\max} & \text{其他} \end{cases} \tag{7-8}$$

$$v = \max \tag{7-9}$$

(2) 由 HSV 到 RGB 的转换公式为

$$h_i = \left[\frac{h}{60}\right] \pmod 6 \tag{7-10}$$

$$f = \frac{h}{60} - h_i \tag{7-11}$$

$$p = v(1 - s) \tag{7-12}$$

$$q = v(1 - f \times s) \tag{7-13}$$

$$t = v(1 - (1 - f) \times s) \tag{7-14}$$

对于 RGB 空间的每一个颜色向量(r, g, b)，则有

$$(r, g, b) = \begin{cases} (v, t, p) & h_i = 0 \\ (q, v, p) & h_i = 1 \\ (p, v, t) & h_i = 2 \\ (p, q, v) & h_i = 3 \\ (t, p, v) & h_i = 4 \\ (v, p, q) & h_i = 5 \end{cases} \tag{7-15}$$

3. YUV 颜色模型

YUV 表示电视信号的彩色坐标系统,当时推出 YUV 模型主要用于优化彩色视频信号的传输,使其向后相容黑白电视。YUV 彩色电视信号传输时,将 R、G、B 变换为亮度信号和色度信号。与 RGB 视频信号传输相比,YUV 的主要优点是占用的频宽较少。PAL 制式将 R、G、B 三色信号改组成 Y、U、V 信号,其中 Y 表示明亮度(luminance 或 luma),也就是灰度值;而 U 和 V 表示的则是色度(chrominance 或 chroma),作用是描述视频帧图像的色彩及饱和度,用于定义像素的颜色。亮度是透过 RGB 输入信号来建立的,色度则定义了颜色的两个方面——色调与饱和度,分别用 u 和 v 来表示。

RGB 与 YUV 之间的转换关系为

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & -0.437 \\ 0.615 & 0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{7-16}$$

若由 YUV 转换到 RGB,则转换关系为

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.395 & -0.581 \\ 1 & 2.032 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \tag{7-17}$$

4. Lab 颜色模型

Lab 颜色模型是国际照明委员会 CIE(commission internationale de l'eclairage)制定和推出的颜色模型,CIE 制定了一系列色度学标准,CIE Lab 颜色体系就是其中应用较为广泛的一项成果。

Lab 模型于 1976 年公布,它也是基于人眼对颜色的视觉感觉而定义的。Lab 模型可以表达正常视力的人能够看到的所有颜色。Lab 描述了颜色的显示方式,而不是设备(如显示器、桌面打印机或数码相机)生成颜色所需的特定色料的数量,因此, Lab 也属于与设备无关的颜色模型。

Lab 色彩模型如图 7-10 所示,由亮度 L 和色彩参数 a、b 三个要素组成。L 表示亮度(luminosity),a 表示从洋红色至绿色的范围,b 表示从黄色至蓝色的范围。Lab 的亮度分量(L)取值范围是 0～100,a、b 的值域均为＋127～－128。当 a＝＋127 时是红色,若 a 渐渐过渡到－128 时,就变成了绿色;b＝＋127 是黄色,过渡到－128 时是蓝色。

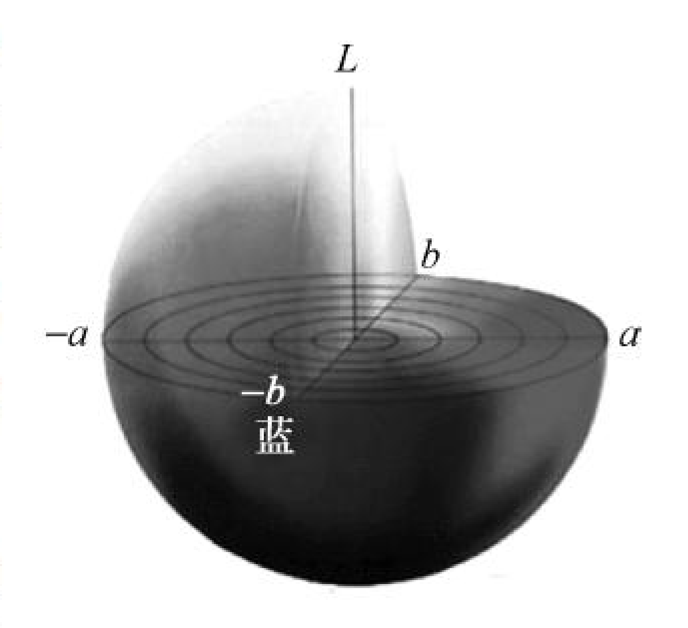


图 7-10 Lab 颜色模型

Lab 色彩模型除不依赖于设备的优点外,还具有一些重要的特点:一是色域宽阔,它不仅包含了 RGB、CMYK 的所有色域,Lab 颜色空间比计算机显示器,甚至比人类视觉的色域范围还要大;二是 Lab 色彩模型改进了 RGB 模型色彩分布不均的缺点,但处理速度却与 RGB 模型同样快。

RGB 模型转换到 Lab 模型的方法如下。

1）由 RGB 转换到 XYZ

$$\begin{cases} R = \text{gamma}\left(\dfrac{r}{255.0}\right) \\ G = \text{gamma}\left(\dfrac{g}{255.0}\right) \\ B = \text{gamma}\left(\dfrac{b}{255.0}\right) \end{cases} \tag{7-18}$$

$$\text{gamma}(x) = \begin{cases} \left(\dfrac{x + 0.055}{1.055}\right)^{2.4} & x > 0.0405 \\ \dfrac{x}{12.92} & \text{其他} \end{cases} \tag{7-19}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{M} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{7-20}$$

其中

$$\boldsymbol{M} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix}$$

上述转换模型中，使用了 X、Y、Z 三刺激值。什么是 X、Y、Z 三刺激值呢？CIE 色度学认为，颜色的感觉是由于 R、G、B 三种原色光刺激的综合结果。在 R、G、B 三原色系统中，红、绿、蓝的刺激量分别以 R、G、B 表示之。由于从实际光谱中选定的、绿、蓝三原色光不能调配出自然界存在的所有色彩。因此，CIE 于 1931 年从理论上假设了并不存在于自然界的三种原色，即理论三原色，以 X、Y、Z 表示，以期从理论上来调(匹)配一切色彩，形成了 XYZ 测色系统。X 原色相当于饱和度比光谱红色还要高的红紫，Y 原色相当于饱和度比 520nm 的光谱绿还要高的绿，Z 原色相当于饱和度比 477nm 的光谱蓝还要高的蓝。这三种理论原色的刺激量以 X、Y、Z 表示，即所谓 X、Y、Z 三刺激值。

2）由 XYZ 转换到 LAB

$$\begin{cases} L^* = 116 \times f\left(\dfrac{y}{y_n}\right) - 16 \\ a^* = 500\left[f\left(\dfrac{x}{x_n}\right) - f\left(\dfrac{y}{y_n}\right)\right] \\ b^* = 200\left[f\left(\dfrac{y}{y_n}\right) - f\left(\dfrac{z}{z_n}\right)\right] \end{cases} \tag{7-21}$$

$$f(t) = \begin{cases} t^{1/3} & t > \left(\dfrac{6}{29}\right)^3 \\ \dfrac{1}{3}\left(\dfrac{29}{6}\right)^2 t + \dfrac{4}{29} & \text{其他} \end{cases} \tag{7-22}$$

上述表达式中，L^*、a^*、b^* 是最终的 Lab 彩色空间三个通道的值，X、Y、Z 一般取值为 95.047、100、108.883。完全按照转换算法运算包含大量的浮点运算，其运算量很大，若用 PC 实现一半没问题，如果在 Android 系统上运行，即使采用 C++进行加速处理，速度问题也需要引起足够的重视。

7.3 伪彩色图像处理

在数字图像处理中还有一种较广泛应用的图像处理技术，即伪彩色增强图像。在 20 世纪 80 年代，由于彩色传感器和彩色图像处理硬件的逐渐成熟，彩色图像处理技术得到了广泛的应用。常用的彩色增强方法有真彩色增强技术、假彩色增强技术和伪彩色增强技术三种。真彩色处理中，被处理的图像一般通过全彩色传感器获取，如彩色摄像机或彩色扫描仪等；假彩色处理是一种尽量逼近真实色彩的人工彩色处理技术；伪彩色是指分配彩色给某一灰度范围或某一灰度值，以增强图像辨识能力。之所以进行彩色增强技术方面的研究，是因为色彩图像具有以下两个重要特点：

(1) 人的视觉系统对色彩非常敏感，人类视觉可以分辨几千种不同的颜色，但却只能分辨出几十种不同的灰度级。即使是经过专门训练的人员（如 X 射线医生）也只能分辨几十种灰度级，而对于彩色来说，人的眼睛可分辨出上千种彩色的色调和强度。因此，在一幅黑白图像中检测不到的信息，经伪彩色增强后可较容易地被检测出来。

(2) 在自动图像分析中，彩色是对分析有利的一个因素，彩色图像通常可使从某些场景中抽取和识别目标的处理变得更加简化。

因此，若能将一幅灰度图像变成彩色图像，就可以在一定程度上达到图像增强的视觉效果。实际生活中许多人并未对假彩色图像处理与伪彩色图像处理进行严格区分，但从图像处理的角度来看，二者还是存在差别的。例如，人们常常把没有颜色的人物照片用人工着色的方法彩色化，其目的主要是使其颜色尽量接近于真实色彩，一般认为这种技术属于假彩色技术。普拉特所提出的假彩色是指将一幅由三基色描述的彩色原始图像或具有同一内容的一套多光谱图像，以线性或非线性方法，逐像素映射到由三激励值所确定的色度空间。

7.3.1 伪彩色处理简介

伪彩色技术在早期的遥感图像处理中已有应用，采用的方法是光学方法。这种方法具有几何失真小的优点，但处理速度非常慢。一般情况下，由于人为设计的各个目标物的颜色是不同的，处理一幅照片需要较复杂的洗印技术，因此在一定程度上限制了其应用范围。

伪彩色处理是图像处理中常用的一种方法，是指通过将每一个灰度级匹配到彩色空间上的一点，将单色图像映射为一幅彩色图像的一种变换，可按某种预先确定的规则生成映射查询表，从而实现给每个灰度级范围赋予一种颜色的彩色图像。因此，伪彩色处理本质上是一种彩色映射过程。一般是将灰度轴匹配到颜色空间中的一条连续的曲线上，例如按某种经验模式进行伪彩色映射的效果通常比随机进行彩色赋值的效果更好。伪彩色处理可以是连续彩色，也可以是由几种颜色构成的彩色。

例 7-1：灰度图像生成伪彩色

如图 7-11 所示是一幅单色灰度图像，通过伪彩色图像增强技术生成的彩色图像如图 7-12所示，该图是通过红色饱和度编码产生的伪彩色图像。

伪彩色增强人为地实现了由灰度图像到彩色图像的变换，从而将人眼难以区分的灰度差异变换为较容易区分的色彩差异。由于原始图像并没有颜色，将其变为彩色的过程实际上是一种被控制的着色过程，所以称为伪彩色增强。常用的伪彩色增强方法有密度分割、伪

彩色变换和频域滤波法三种。


图 7-11　灰度图像


图 7-12　伪彩色增强图像

7.3.2　密度分割

密度分割是伪彩色处理技术中最简单而常用的一种方法。其基本原理是将单色图像的灰度值动态范围分割为多个灰度值区间，使分割后的每一区间对应某一种颜色，必要时可以使每一灰度值对应一种颜色，形成伪彩色图像。

灰度图像可以视为二维亮度函数，用函数 $f(x,y)$ 表示一幅灰度图像，如图 7-13 所示，分别用平行于图像 xOy 坐标平面的 L_1、L_2 平面去切割图像函数 $f(x,y)$，这样函数 $f(x,y)$ 在分割处 L_1、L_2 被分为了三个灰度值区间，形成三个部分。

分割完成以后，如果再对每一个区间赋予某种颜色，就可以将原来的灰度图像变换成具有三种颜色的彩色图像。如图 7-14 所示，以 $g(x,y)$ 表示变换后的彩色图像，即目标图像，为表示方便，图像 $f(x,y)$ 切割处的像素值用切割平面符号表示。如果用 N 个平面 L_1、L_2、…、L_N 去切割图像函数 $f(x,y)$，则可以得到 $N+1$ 个灰度值区间，每一个区间对应一种颜色 C_N。于是，对于每一像素点 (x,y) 就可以按图 7-14 所示方式形成颜色匹配关系。

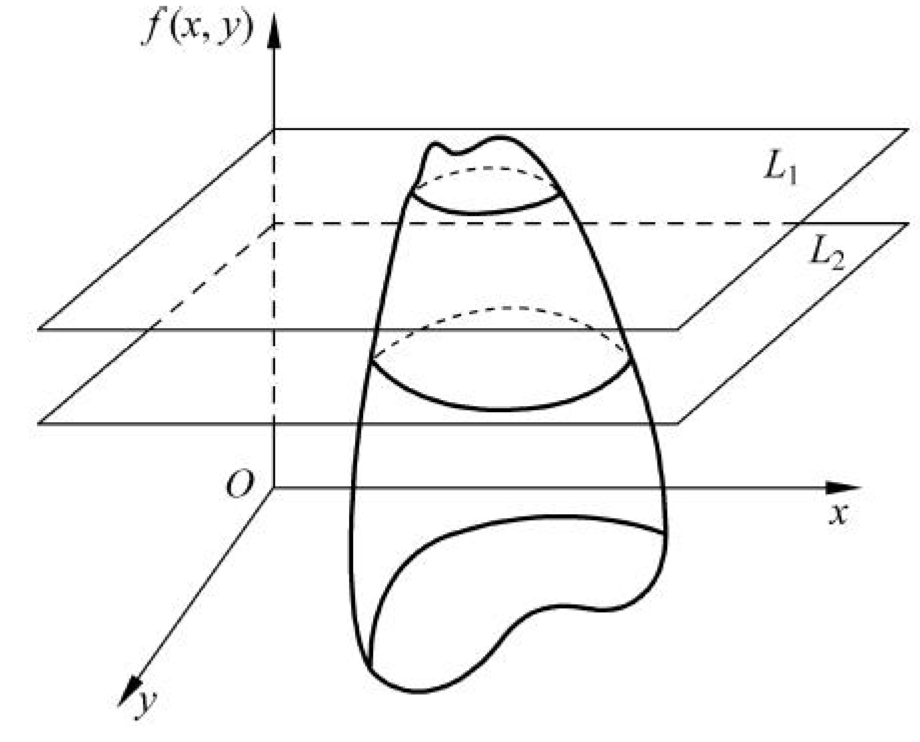


图 7-13　密度分割示意图

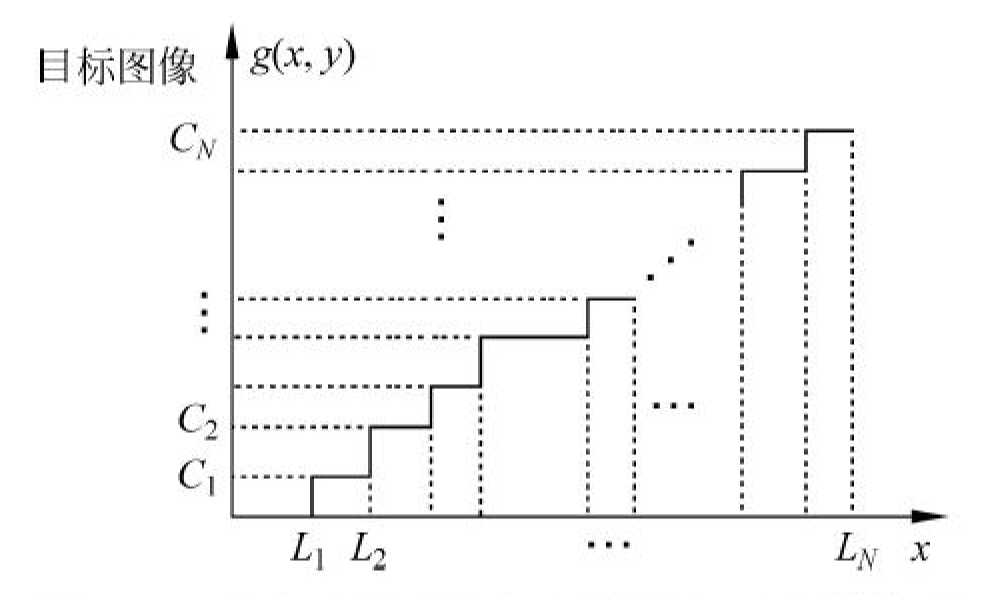


图 7-14　多灰度伪彩色切割图颜色匹配梯关系

如果将每一个灰度值都划分成一个区间，如将 8bit 灰度图像划分成 256 个区间，则可以形成索引图像，因此，可以认为索引图像是由灰度图像经最大密度分割而生成的图像。需

要注意的是,每一灰度值区间赋予何种颜色,是由具体应用所决定的,并无规律可言。但总的来讲,相邻灰度值区间的颜色差别不宜太小也不宜太大,太小将无法反映细节上的差异,太大则会导致图像出现不连续性。实际应用中,密度切割平面之间可以是等间隔的,也可以是不等间隔的,而且切割平面的划分也应依据具体的应用范围和研究对象而确定。

基于密度分割的伪彩色处理的特点是简单易行,便于用软件或硬件实现,可以实现实时处理,而且具有广泛的用途和很高的精度。

7.3.3 灰度级伪彩色变换

密度分割法实质上是通过一个分段线性函数实现从灰度到彩色的变换,每个像素点只经过一个变换对应到某一种颜色。灰度级伪彩色变换与密度分割有所不同,其基本原理是,将输入图像的每一像素点的灰度值进行三个互相独立的变换,分别产生红、绿、蓝三个分量图像,然后将其合成为一幅彩色图像。

伪彩色变换过程如图 7-15 所示,将灰度图像送入具有不同变换特性的红、绿、蓝 3 个变换器,再将 3 个变换器的不同输出分别送到彩色显像管的红、绿、蓝电子枪。同一灰度由 3 个变换器对其实施不同变换,根据 3 个变换器的不同输出,在彩色显像管内合成某种彩色。可见,不同的灰度级一定可以合成不同色彩。灰度级伪彩色变换可以将灰度图像变为具有多种颜色渐变的连续彩色图像,这种伪彩色处理技术在遥感技术中常称为假彩色合成方法。

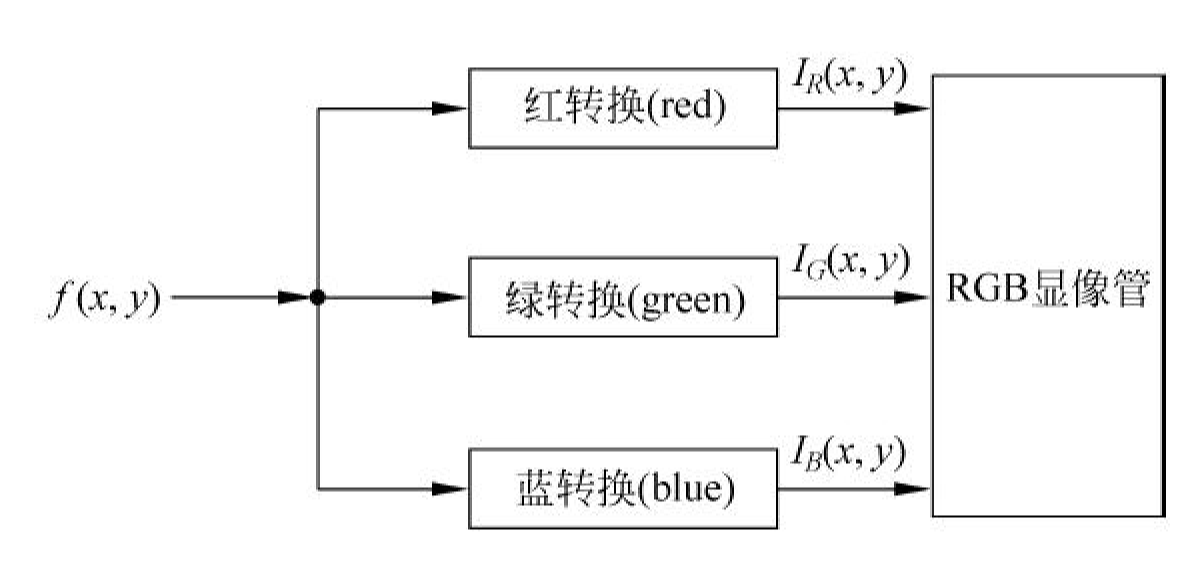


图 7-15 伪彩色变换原理

变换公式的形式为

$$\begin{cases} R(x,y) = T_R[f(x,y)] \\ G(x,y) = T_G[f(x,y)] \\ B(x,y) = T_B[f(x,y)] \end{cases} \tag{7-23}$$

式中,$f(x,y)$表示原始图像灰度值函数;$R(x,y)$、$G(x,y)$、$B(x,y)$分别是伪彩色中红、绿、蓝三基色分量的数值;$T_r[]$、$T_G[]$、$T_B[]$分别表示与原始图像灰度值的函数关系。

三个变换函数要求相互独立,但在实际应用中变换函数可以取带绝对值的正弦函数、线性变换函数等同一类型的函数。图 7-16 给出了一组经典的连续变换函数,灰度值范围为$[0,L]$,每个变换取不同的分段线性函数。可以看出,最小的灰度值 0 映射为蓝色,中间的灰度值 $L/2$ 映射为绿色,最高的灰度值 L 映射为红色,其余的灰度值根据该曲线分别映射为相应的颜色。

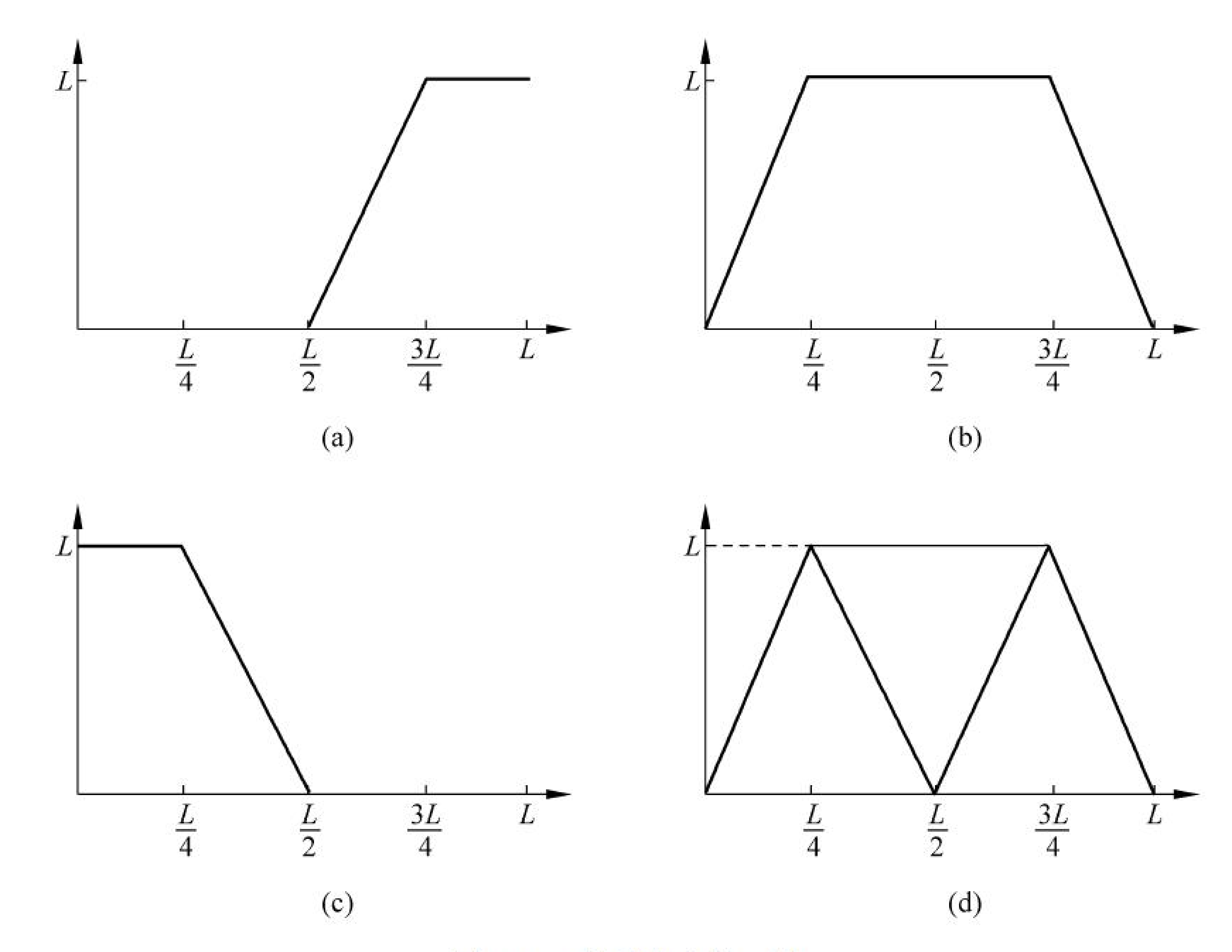


图 7-16 伪彩色变换函数

7.4 全彩色图像处理

自然界的颜色是多种多样、丰富多彩的，能真实反映自然物体真实色彩的图像叫真彩色图像或全彩色图像，若彩色图像的单通道位深度为 8 位，则真彩色图像是指 24 位 RGB 图像。RGB 图像一般由彩色照相机、彩色摄像机或其他彩色数字化设备获取。

7.4.1 彩色图像处理方法分类

RGB 真彩色图像的处理一般可以分为合成处理法和直接处理法两大类。

（1）合成处理法，即将 RGB 图像分解为 R、G、B 三个波段的图像分量，分别对 R 分量、G 分量、B 分量的图像进行处理，处理之后合成处理后的彩色图像。

（2）直接处理法，指直接处理每一个彩色像素，即每一个像素是一个三维分量，令 c 表示 RGB 图像任意像素的向量，则有

$$c=\begin{bmatrix}C_R\\C_G\\C_B\end{bmatrix}=\begin{bmatrix}R\\G\\B\end{bmatrix} \tag{7-24}$$

一幅大小 $M\times N$ 的图像，则有 MN 个 24 分量的彩色向量 $c(x,y)$，即

$$c(x,y)=\begin{bmatrix}C_R(x,y)\\C_G(x,y)\\C_B(x,y)\end{bmatrix}=\begin{bmatrix}R(x,y)\\G(x,y)\\B(x,y)\end{bmatrix} \tag{7-25}$$

$$x=1,2,3,\cdots,M-1;\ y=1,2,3,\cdots,N-1$$

这两种方法都应用了多幅灰度图像的合成处理技术，在计算机中，任何一种颜色都可以用红、绿、蓝三基色通过合成得到，因此这两种技术一般是将三幅图像分别作为红、绿、蓝三个通道进行合成。

7.4.2 彩色图像直方图增强

从前述章节中可以看到，直方图处理对于灰度图像具有较好增强效果，同样，直方图处理对于彩色图像也具有很好的效果。彩色图像直方图均衡化的作用是扩展输入图像各分量灰度值的动态范围，提高图像的对比度和亮度。

例 7-2：彩色图像直方图增强实验

本实验观察彩色图像直方图增强处理效果。图 7-17(a)为原始图像，这是一幅广泛使用的 lena.jpg 图像，原图像的 R 分量、G 分量、B 分量的图像分别如图 7-17(b)、(c)、(d)所示。将原始图像的 R、G、B 分量图像分别进行直方图均衡化处理，以增强图像的对比度，然后再由处理后的三个分量矩阵合成 RGB 图像。



(a) 原始图像　(b) R分量图像　(c) G分量图像　(d) B分量图像

图 7-17　RGB 原始图像及各分量图像

这里选择 lena 图像作为彩色图像处理测试图像，不是因为 lena 漂亮，而是因为该图像具有相对复杂的细节和纹理，同时也有大片的平滑区域，适度地混合了细节、平滑区域、阴影和纹理多方面的综合信息，从而能很好地测试各种图像处理算法。彩色图像 lena 均衡化以后的结果图像如图 7-18 所示。本实验结果表明，直方图均衡化对彩色图像的显示效果也有

显著的增强作用。对比度增强源于直方图在整个亮度标度上的显著扩展，亮度的增强是因为均衡化后的图像的直方图中灰度级平均值高于原始值。

7.4.3 彩色图像去噪

彩色图像去噪基础理论和灰度图像去噪方法类似，不同之处是需按 R、G、B 三分量分别进行滤波消噪，在彩色图像消噪中，前述章节介绍过的空域滤波和频域滤波方法都可以根据情况选择采用。例 7-3 是彩色图像中值滤波消除噪声实验。

例 7-3：彩色图像噪声处理实验

本例介绍了通过中值滤波消除噪声的方法和步骤，选择以图 7-17(a)所示 lena.jpg 照片作为原始图像，给该图像添加方差为 0.05 的高斯噪声，添加噪声后的 lena 图像如图 7-19 所示。



图 7-18 RGB 均衡化以后的图像



图 7-19 带噪声的 lena 图像

在大的步骤上，彩色图像消噪方法的步骤和直方图增强方法的步骤基本类似，首先将源图像的 R、G、B 分量数据分别读出，再对三个分量灰度矩阵分别进行滤波，结合噪声和图像的特点，采用中值滤波消除噪声，完成滤波之后按 R、G、B 合成新的彩色图像，滤波结果如图 7-20 所示。从中可以看出，噪声得到了明显的抑制，消噪效果非常明显。



图 7-20 消除噪声后的 lena 图像

MATLAB 代码如下：

```
clc;
clear all;
Image_a = imread('C:\Users\chen\Desktop\color_lena1.
jpg');
j = imnoise(Image_a,'gaussian',0.05);
figure(1);
imshow(Image_a);
figure(2);
subplot(1,2,1);imshow(j);
```

```
b = j(:,:,1);
c = j(:,:,2);
d = j(:,:,3);

b1 = medfilt2(b);
c1 = medfilt2(c);
d1 = medfilt2(d);
Image_a1(:,:,1) = b1;
Image_a1(:,:,2) = c1;
Image_a1(:,:,3) = d1;
subplot(1,2,2);imshow(Image_a1);
```

7.4.4 彩色频域处理

与灰度图像一样,彩色图像处理也可以在频域进行。彩色图像频域滤波的基本原理是,先将 RGB 彩色图像的 R、G、B 三个分量分别进行傅里叶变换,并根据各分量的特点和图像处理的目的,对各分量分别选择进行低通、高通、带通或带阻滤波,以及对各分量信息进行其他处理。例如,为了突出高频信息将其转换为红色,只需在红色通道滤波器设计为具有高通特性即可,然后再对各分量进行 IFFT 变换,最后合成彩色图像,完成全彩色处理。频域滤波的过程如图 7-21 所示。

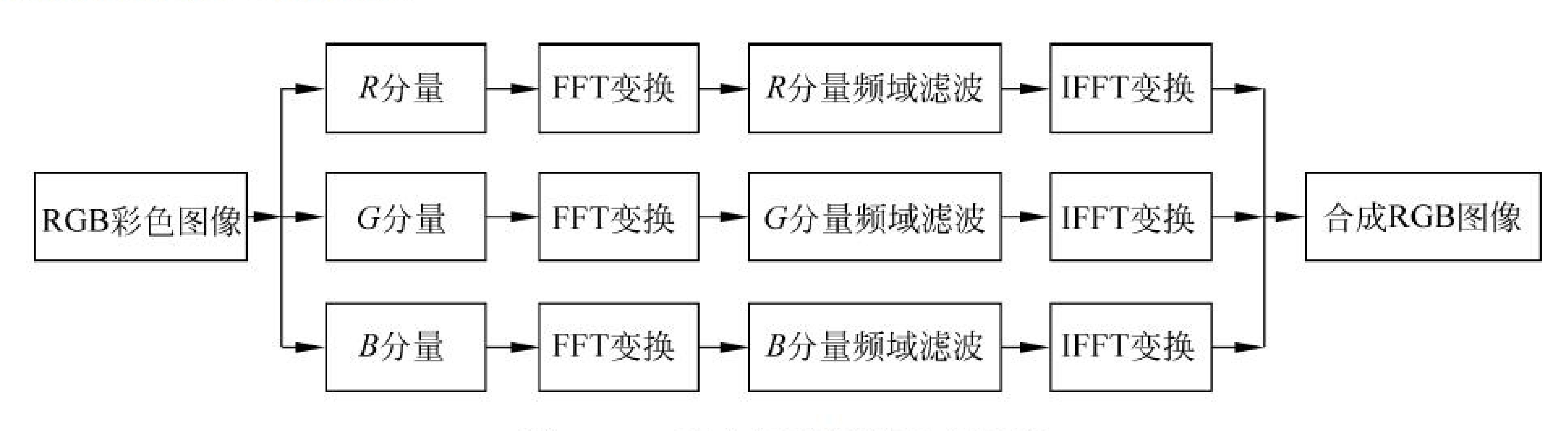


图 7-21 彩色图像频域处理过程

习题

1. 什么是索引图像和 RGB 彩色图像?两者有何区别和联系?
2. 简述人眼视觉模型的基本原理。
3. 在电磁波谱中,可见光的波长范围是多少?
4. 常用的计算机的颜色模型有哪几种类型?各有何特点?
5. 面向硬件设备的颜色模型主要有哪几种?
6. 面向视觉感知的颜色模型主要有哪几种?
7. RGB 与 HSI 颜色模型各有何特点?如何实现数字图像的 RGB 到 HSI 模型的转换?
8. 编写程序观察一幅彩色图像的 R、G、B 分量图像。
9. 编写程序观察一幅彩色图像的 H、S、I 分量图像。

10. 编写 MATLAB 程序实现由 RGB 到 HSV 颜色体系的转换。
11. 编写程序观察一幅彩色图像的 H、S、V 分量图像。
12. 编写 MATLAB 程序实现由 RGB 到 YUV 颜色体系的转换。
13. 简要阐述如何实现伪彩色图像增强。
14. 简述彩色图像增强的基本原理和步骤。
15. 选择一幅彩色图像,编写 MATLAB 程序实现图像的直方图均衡化。

第8章 图像复原

CHAPTER 8

图像复原又称为图像恢复，最初主要是针对图像在成像过程中的“退化”而提出来的，而成像过程中的“退化”现象主要指成像系统受到各种因素的影响，如成像系统的散焦、设备与物体间存在相对运动或器材的固有缺陷等，导致图像的质量不能够达到理想要求。因此，需要研究图像复原技术以便对图像进行恢复。图像复原与图像增强技术一样，也是一种改善图像质量的技术方法。图像复原的基本方法是根据事先建立起来的成像系统退化模型，将退化了的图像以最大的保真度恢复为原始图像或物体的真实影像。本章围绕图像退化的原因，首先讨论图像退化数学模型的概念，然后讨论图像的无约束复原及逆滤波复原、有约束复原、匀速运动模糊恢复等方法的基本原理，最后介绍几何畸变复原、盲目图像复原及中值滤波复原方法等。

8.1 图像退化机理

在数字图像的获取过程中，由于各种各样的原因，图像会产生一定程度的退化。图像获取过程中因光学成像系统的像差、光学成像衍射、成像系统的非线性畸变、摄影胶片感光特性的非线性，以及成像过程的相对运动、传输介质、大气湍流和环境因素等原因，都会不同程度地造成图像质量的下降或退化。

8.1.1 退化原因

造成图像退化的原因很多，一般认为以下几种情况可导致图像质量的退化：

(1) 成像系统镜头聚焦不准产生的散焦。

(2) 相机与景物之间的相对运动。

(3) 成像系统存在的各种非线性因素以及系统本身的性能等因素使图像质量下降。

(4) 遥感遥测中，由于飞机、轨道卫星的运动稳定性，以及地球自转等因素引起的遥感图像几何失真。

(5) 模拟图像在数字化过程中，因数字化的精度和误差而不同程度地损失一些图像细节。

(6) 射线辐射、大气湍流等因素造成的照片畸变。

(7) 成像系统的像差、畸变、有限带宽等非理想因素造成的图像失真。

(8) 底片感光、图像显示时会造成记录显示失真。

(9) 成像系统中存在的各种随机噪声。

8.1.2 复原机理

图像复原的过程一般是沿着图像退化的逆向过程进行的，因此，首先根据先验知识分析退化原因，了解图像变质的机理，在此基础上建立图像的退化模型，然后以相反的过程(退化的逆过程)对图像进行处理，使图像质量得到补偿和改善。通常情况下，图像复原的质量是根据一些规定的客观准则来评价的，如最小均方准则、加权均方准则等，也可以采用定性标准，即图像接近全真景物图像的程度是否到达最佳。

虽然图像复原与图像增强从表面上看都是为了提高图像的质量，但它们之间无论是过程还是目的都存在明显的区别。

图像增强技术是采用特定技术来突出和强调图像中所关注的特征，在对图像进行增强处理的过程中，一般不考虑图像退化的真实物理过程，只关注图像增强后的实际效果，根据目的和用途的不同，既可以增强一幅图像的全部，也可以仅增强图像的某一部分；与原始图像相比，增强后的图像还可能损失掉部分信息。因此，图像增强的目的是提高视感质量，但增强后的图像可能与原始图像有一定的差异。

图像复原是直接针对图像产生退化的原因，建立相应的数学模型，采取沿着图像降质的逆过程等技术手段对退化图像进行有针对性的补偿，使恢复后的图像最大限度地接近原始图像。因此，图像复原不是为了突出或增强图像的某些信息，而是为了减少或消除各种因素所造成的图像质量下降，恢复图像的本来面目。也就是说，使待复原的图像与原始图像尽可能一致。图像复原的主要任务是建立图像复原的反向过程的数学模型。

由于导致图像退化的原因众多、性质各不相同，相应地描述图像退化过程的数学模型也多种多样，而恢复的质量标准也往往存在差异性。因此，图像复原是一个复杂的数学过程，退化原因不同，图像复原的方法、技术也各不相同。

8.2 图像退化模型

图像复原技术的主要任务是建立系统的退化模型。设 $f(x,y)$为输入图像，$f(x,y)$经过某个退化系统后输出的是一幅退化的图像，退化系统以 $H(x,y)$表示，退化过程中，噪声引起的退化一般可视为加性噪声，虽然从噪声的种类而言还包括乘性噪声，但从图像系统的应用情况看，无论量化噪声，还是随机噪声等都普遍表现为加性噪声，噪声用 $n(x,y)$表示。

因此，在考虑噪声的影响以后，图像 $f(x,y)$的退化过程表现为，$f(x,y)$经过一个退化系统 $H(x,y)$的作用，并与噪声 $n(x,y)$叠加，形成退化后的图像 $g(x,y)$。这一过程的输入和输出关系如图 8-1 所示。

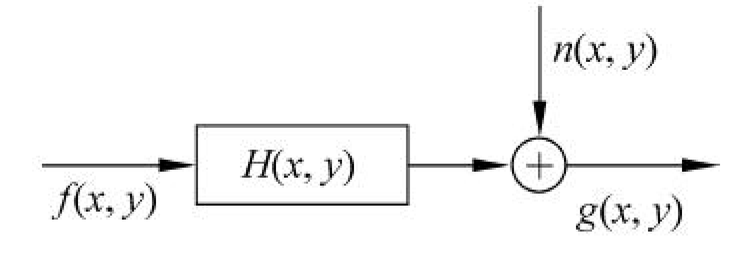


图 8-1 图像退化系统的数学模型

该退化模型中，$H(x,y)$概括了图像退化系统的物理过程，是研究退化所欲探求的数学模型。图像复原问题就是根据已有的退化图像 $g(x,y)$，通过寻求合理的退化系统 $H(x,y)$，以退化的逆过

程去获取原始图像 $f(x,y)$，或者说是逆向地寻找原始图像的最佳估计。

根据上述分析和图 8-1 所示模型，如果系统没有噪声或者不考虑噪声的影响，图像退化的过程可以用数学表达式写为

$$g(x,y)=H\cdot f(x,y) \tag{8-1}$$

考虑系统噪声的影响时，退化模型为

$$g(x,y)=H\cdot f(x,y)+n(x,y) \tag{8-2}$$

式中，H 表示 $H(x,y)$，即将图像的退化过程抽象为一个系统(或称算子)H 的作用；噪声 $n(x,y)$是一种统计性质的信息，一般可用白噪声模型表示，白噪声的一般特性是频谱密度为常数，并与图像不相关。

虽然从严格意义上讲，一切图像系统都表现出非线性、时变等特性，但实际应用中，根据人类视觉的灵敏性以及实际图像系统的近似特性，实际图像系统都可以用线性、时不变和空间不变的系统模型近似表示。这一点不仅给图像处理和图像复原工作带来了方便，也比非线性系统更具有实际意义和可用性。

8.2.1 退化模型的一般特性

由于实际图像系统及其退化系统都表现为线性、时不变和位置不变特性，因此，退化系统 H 具有以下两个主要性质。

1. 线性特性

若 k_1、k_2 为常数，当不考虑噪声作用，即 $n(x,y)=0$ 时，则

$$\begin{aligned}H[k_1f_1(x,y)+k_2f_2(x,y)]&=H[k_1f_1(x,y)]+H[k_2f_2(x,y)]\\&=k_1g_1(x,y)+k_2g_2(x,y)\end{aligned}$$

(1) $k_1=k_2=1$ 时，表现为可加性，即

$$\begin{aligned}H[f_1(x,y)+f_2(x,y)]&=H[f_1(x,y)]+H[f_2(x,y)]\\&=g_1(x,y)+g_2(x,y)\end{aligned}$$

(2) k_1、k_2 任一个常数为 0 时(如 $k_2=0$)，表现为比例性，即

$$H[k_1f_1(x,y)=k_1g_1(x,y)$$

2. 空间位置不变性

对任一图像 $f(x,y)$和常数 α、β，若

$$g(x,y)=H\cdot f(x,y)$$

则

$$H\cdot f(x-\alpha,y-\beta)=g(x-\alpha,y-\beta)$$

空间位置不变性的实际意义是，对图像 $f(x,y)$上任一点的处理结果，只取决于该点的输入值，而与该点的坐标位置无关。

8.2.2 连续退化模型

下面介绍连续图像退化的数学模型。

设 $f(x,y)$表示连续图像，根据 δ 冲激函数的筛选性质，则连续图像 $f(x,y)$可以表示为 δ 函数与 $f(x,y)$的卷积积分，即

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(\alpha,\beta)\delta(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta$$

δ 函数的物理意义表示空间上的点脉冲。

一般情况下，不考虑噪声的影响(设退化模型中的 $n(x,y)=0$)，根据 δ 函数的性质，连续图像经过退化系统 H 后的输出为

$$g(x,y)=H\cdot f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}H[f(\alpha,\beta)\delta(x-\alpha,y-\beta)]\mathrm{d}\alpha\mathrm{d}\beta$$

由于 $f(\alpha,\beta)$ 与 x、y 无关，根据线性齐次性可得

$$\begin{aligned}g(x,y)&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(\alpha,\beta)H\delta(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta\\&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(\alpha,\beta)h(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta\end{aligned}\tag{8-3}$$

式中，$h(x-\alpha,y-\beta)$ 可以表示为 $h(x,\alpha,y,\beta)$，称为系统的冲激响应函数，它表示系统 H 对坐标为 (α,β) 处的冲激函数 $\delta(x-\alpha,y-\beta)$ 的响应。在光学中，冲激为一个光点，因此，$h(x,\alpha,y,\beta)$ 一般也被称为点扩散函数。

根据式(8-3)可知，若系统的冲激响应函数已知，就可以知道退化系统是如何形成的，即对任一输入 $f(\alpha,\beta)$，其响应都可以通过式(8-3)进行卷积计算，即

$$g(x,y)=f(x,y)*h(x,y)\tag{8-4}$$

由于图像的退化过程可以认为是线性空间的不变系统，因此系统输出的退化图像 $g(x,y)$ 应为输入图像和系统冲激响应的卷积积分。

对式(8-4)两边进行傅里叶变换，可得

$$G(u,v)=H(u,v)F(u,v)\tag{8-5}$$

式中，$G(u,v)$、$F(u,v)$ 分别是 $g(x,y)$、$f(x,y)$ 的二维傅里叶变换；函数 $H(u,v)$ 称为退化系统的传递函数，它是退化系统冲激响应 $h(x,y)$ 的傅里叶变换。

若考虑加性噪声 $n(x,y)$ 的影响，则连续函数的退化模型可表示为

$$g(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(\alpha,\beta)h(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta+n(x,y)\tag{8-6}$$

即

$$g(x,y)=f(x,y)*h(x,y)+n(x,y)\tag{8-7}$$

同样，对上式两边进行傅里叶变换，可得

$$G(u,v)=H(u,v)F(u,v)+N(u,v)\tag{8-8}$$

式中，$N(u,v)$ 为噪声函数 $n(x,y)$ 的傅里叶变换。

由于采用了线性模型，大多数情况下，图像复原的关键任务是求解冲激响应函数 $h(x,y)$。一般而言，系统的传递函数 $H(u,v)$ 比较容易获得，而 $H(u,v)$ 是 $h(x,y)$ 的傅里叶变换，因此，图像复原所研究的重点是线性退化模型。

8.2.3 离散退化模型

数字图像讨论的是离散的图像函数，因此需要对连续模型进行离散化处理，即将连续模型中的积分以求和的形式表示。

1. 一维离散退化模型

不考虑噪声的情况下，设 $f(x)$ 被均匀采样后形成具有 A 个采样值的离散输入函数，$h(x)$ 被采样后形成具有 B 个采样值的退化系统冲激响应。因此，连续函数退化模型中的连

续卷积关系就演变为离散卷积和关系。

如果 $f(x)$ 和 $h(x)$ 都是具有周期为 N 的序列，那么离散的退化模型可表示为

$$\begin{aligned} g(x) &= f(x) * h(x) \\ &= \sum_m f(m)h(x-m) \end{aligned} \tag{8-9}$$

显然，$g(x)$ 也是具有周期 N 的序列，周期卷积可用常规卷积法计算。

如果 $f(x)$ 和 $h(x)$ 均不具有周期性，则采用在末端补零的方法延伸，使其扩展为周期函数 $f_e(x)$ 和 $h_e(x)$，令周期 $M=A+B-1$，则 $f_e(x)$ 和 $h_e(x)$ 可分别表示为

$$f_e(x) = \begin{cases} f(x) & 0 \leqslant x \leqslant A-1 \\ 0 & A-1 < x \leqslant M-1 \end{cases}$$

$$h_e(x) = \begin{cases} h(x) & 0 \leqslant x \leqslant B-1 \\ 0 & B-1 < x \leqslant M-1 \end{cases}$$

这样，可得到如下离散卷积退化模型：

$$g_e(x) = \sum_m f_e(m)h_e(x-m) \tag{8-10}$$

式中，$x=0,1,2,\cdots,M-1$。显然，$g_e(x)$ 也是周期为 M 的函数。

式(8-10)也可以用如下矩阵形式表示：

$$\begin{bmatrix} g_e(0) \\ g_e(1) \\ g_e(2) \\ \cdots \\ g_e(M-1) \end{bmatrix} = \begin{bmatrix} h_e(0) & h_e(-1) & \cdots & h_e(-M+1) \\ h_e(1) & h_e(0) & \cdots & h_e(-M+2) \\ h_e(2) & h_e(1) & \cdots & h_e(-M+3) \\ \cdots & \cdots & \cdots & \cdots \\ h_e(M-1) & h_e(-2) & \cdots & h_e(0) \end{bmatrix} \begin{bmatrix} f_e(0) \\ f_e(1) \\ f_e(2) \\ \cdots \\ f_e(M-1) \end{bmatrix} \tag{8-11}$$

上式可以写为矩阵方程式，即

$$\boldsymbol{g} = \boldsymbol{H} \cdot \boldsymbol{f} \tag{8-12}$$

式中

$$\boldsymbol{f} = [f_e(0) \quad f_e(1) \quad f_e(2) \quad \cdots \quad f_e(M-1)]^{\mathrm{T}}$$

$$\boldsymbol{g} = [g_e(0) \quad g_e(1) \quad g_e(2) \quad \cdots \quad g_e(M-1)]^{\mathrm{T}}$$

$$\boldsymbol{H} = \begin{bmatrix} h_e(0) & h_e(-1) & \cdots & h_e(-M+1) \\ h_e(1) & h_e(0) & \cdots & h_e(-M+2) \\ h_e(2) & h_e(1) & \cdots & h_e(-M+3) \\ \cdots & \cdots & \cdots & \cdots \\ h_e(M-1) & h_e(-2) & \cdots & h_e(0) \end{bmatrix}$$

$\boldsymbol{g}$、$\boldsymbol{f}$ 分别是 $M\times 1$ 列向量，$\boldsymbol{H}$ 为 $M\times M$ 循环矩阵。由于 $h_e(x)$ 的周期为 M，因此下列表达式成立。

$$\begin{cases} h_e(-1) = h_e(M-1) \\ h_e(-2) = h_e(M-2) \\ \cdots \\ h_e(-M+2) = h_e(2) \\ h_e(-M+1) = h_e(1) \end{cases}$$

于是，$\boldsymbol{H}$ 矩阵可以简化为

$$\boldsymbol{H}=\begin{bmatrix} h_e(0) & h_e(-1) & \cdots & h_e(1) \\ h_e(1) & h_e(0) & \cdots & h_e(2) \\ h_e(2) & h_e(1) & \cdots & h_e(3) \\ \cdots & \cdots & \cdots & \cdots \\ h_e(M-1) & h_e(-2) & \cdots & h_e(0) \end{bmatrix}$$

2. 二维离散退化模型

设二维数字图像用 $f(x,y)$ 表示，因此，二维输入图像 $f(x,y)$ 和冲激响应 $h(x,y)$ 分别具有 $A\times B$ 和 $C\times D$ 个元素，为避免交叠误差，与一维离散模型类似，采用末端补零的方法延拓，将它们扩展成 $M\times N$ 个元素的周期扩展图像，其中 $M=A+C-1$，$N=B+D-1$，即

$$f_e(x,y)=\begin{cases} f(x,y) & 0\leqslant x\leqslant A-1, 0\leqslant y\leqslant B-1 \\ 0 & A-1<x\leqslant M-1, B-1<y\leqslant N-1 \end{cases}$$

$$h_e(x,y)=\begin{cases} h(x,y) & 0\leqslant x\leqslant C-1, 0\leqslant y\leqslant D-1 \\ 0 & C-1<x\leqslant M-1, D-1<y\leqslant N-1 \end{cases}$$

延拓后，$f_e(x,y)$ 和 $h_e(x,y)$ 分别成为二维周期函数，在 x 和 y 方向上的周期分别为 M 和 N，与一维离散系统类似，则输出的二维退化数字图像为

$$g_e(x,y)=\sum_m\sum_n f_e(m,n)h_e(x-m,y-n) \tag{8-13}$$

在该离散卷积退化模型中没有考虑噪声项，若考虑噪声的作用，只需在式(8-13)后加上一个 $M\times N$ 的扩展离散噪声项 $n(x,y)$，即可得到完整的二维离散退化模型：

$$g_e(x,y)=\sum_m\sum_n f_e(m,n)h_e(x-m,y-n)+n(x,y) \tag{8-14}$$

式中，$x=0,1,2,\cdots,M-1$； $y=0,1,2,\cdots,N-1$。

与一维离散模型类似，二维离散退化模型也可用矩阵来表示：

$$\boldsymbol{g}=\boldsymbol{H}\cdot\boldsymbol{f}$$

式中，$\boldsymbol{g}$ 和 $\boldsymbol{f}$ 代表 MN 维列向量，即 $MN\times 1$ 维向量，这些列向量是由 $M\times N$ 维的函数矩阵 $f(x,y)$、$g(x,y)$ 的行向量一次组成为一维列向量，形式分别为

$$\boldsymbol{f}=[f_e(0,0)\cdots f_e(0,N-1)f_e(1,0)\cdots f_e(1,N-1)\cdots f_e(M-1,0)\cdots f_e(M-1,N-1)^{\mathrm{T}}]$$

$$\boldsymbol{g}=[g_e(0,0)\cdots g_e(0,M-1)g_e(1,0)\cdots g_e(1,M-1)\cdots g_e(M-1,0)\cdots g_e(M-1,N-1)^{\mathrm{T}}]$$

式中，$\boldsymbol{H}$ 是 $MN\times MN$ 矩阵，矩阵 $\boldsymbol{H}$ 可表示为如下 $M\times M$ 分块循环矩阵的形式，即

$$\boldsymbol{H}=\begin{bmatrix} [H_0] & [H_{M-1}] & [H_{M-2}] & \cdots & [H_1] \\ [H_1] & [H_0] & [H_{M-1}] & \cdots & [H_2] \\ [H_2] & [H_1] & [H_0] & \cdots & [H_3] \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ [H_{M-1}] & [H_{M-2}] & [H_{M-3}] & \cdots & [H_0] \end{bmatrix}$$

其中每一个分块矩阵 $\boldsymbol{H}_j$ 是 $N\times N$ 的矩阵，形式为

$$\boldsymbol{H}_j=\begin{bmatrix} h_e(j,0) & h_e(j,N-1) & \cdots & h_e(j,1) \\ h_e(j,1) & h_e(j,0) & \cdots & h_e(j,2) \\ h_e(j,2) & h_e(j,1) & \cdots & h_e(j,3) \\ \cdots & \cdots & \cdots & \cdots \\ h_e(j,N-1) & h_e(j,N-2) & \cdots & h_e(j,0) \end{bmatrix}$$

若考虑噪声的作用，则二维离散退化模型可表示为

$$\boldsymbol{g}=\boldsymbol{H}\cdot\boldsymbol{f}+n \tag{8-15}$$

图像复原的问题即获取原图像 $f(x,y)$，因此，在线性和空间位置不变的条件下，图像复原问题可归结为获得退化系统的冲激响应 $h(x,y)$和加性噪声 $n(x,y)$，然后，根据 $\boldsymbol{g}=\boldsymbol{H}\cdot\boldsymbol{f}+n$，就可以估计出理想图像 $f(x,y)$。

一个不容忽视的问题是，对于一般的图像复原，其计算量是非常大的。以一般大小的数字图像为例，如 $M=N=512$，则矩阵 $\boldsymbol{H}$ 的大小为$M^2\times M^2=262\ 144\times 262\ 144$。可见，为了计算得到 $\boldsymbol{f}$，需求解 262 144 个线性方程组，其计算量之大是不难想象的。降低计算量的主要方法是应充分利用循环矩阵 $\boldsymbol{H}$ 的有关性质。

8.3 无约束复原

在已经给定退化图像 $g(x,y)$的条件下，图像复原完全取决于对退化系统 $h(x,y)$及噪声 $n(x,y)$的了解，从而估计出原始图像 $f(x,y)$。实际上，为了保证估计的精度和准确性，这类估计应在某种最佳准则下达到最优性质。例如，在均方误差最小意义下，原始图像 $f(x,y)$的最佳估计。

目前，包括最大绝对误差最小、平均绝对误差最小或互相关为最大等在内的各种估计准则中，相对而言，均方误差最小估计是最简单易行的方法，并由此可以导出其他较实用的复原方法。

8.3.1 无约束复原基本原理

根据系统退化模型，噪声项可以表示为

$$n=\boldsymbol{g}-\boldsymbol{H}\cdot\boldsymbol{f}$$

在噪声 n 不确定的情况下，希望对原始图像 $\boldsymbol{f}$ 的估计值$\hat{\boldsymbol{f}}$应满足这样的条件，使 $\boldsymbol{H}\hat{\boldsymbol{f}}$在最小二乘方误差意义上近似于 g。也就是说，希望找到一个$\hat{\boldsymbol{f}}$，使得

$$\|n\|^2=\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2=(\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}})^{\mathrm{T}}(\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}) \tag{8-16}$$

为最小(噪声项的范数最小)。由定义可知

$$\|n\|^2=n^{\mathrm{T}}n$$

求$\|n\|^2$ 最小等效于求$\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2$ 最小，即

$$J(\hat{\boldsymbol{f}})=\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2 \tag{8-17}$$

在这里，除了要求估计的$\hat{\boldsymbol{f}}$使 $J(\hat{\boldsymbol{f}})$为最小值以外，不受任何其他条件的约束，因此称为无约束复原方法。

使 $J(\hat{f})$为最小的条件是 $J(\hat{f})$对$\hat{f}$的微分为 0,即

$$\frac{\partial}{\partial \hat{f}} J(\hat{f}) = -2\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{g} - \boldsymbol{H}\hat{\boldsymbol{f}}) = 0 \tag{8-18}$$

因此,可得

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\hat{\boldsymbol{f}} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \tag{8-19}$$

即

$$\hat{\boldsymbol{f}} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \tag{8-20}$$

如果 x 和 y 方向上的采样点数相同,即 $M=N$,$\boldsymbol{H}$ 为方阵且 $\boldsymbol{H}$ 的逆矩阵存在,则

$$\hat{\boldsymbol{f}} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} = \boldsymbol{H}^{-1}\boldsymbol{g} \tag{8-21}$$

上式表明,对退化图像的复原估计中,若退化系统 $\boldsymbol{H}$ 已知,就可以根据式(8-20)求解出原始图像 $\boldsymbol{f}$ 的均方误差最小估计值$\hat{\boldsymbol{f}}$。若 $\boldsymbol{H}$ 的逆矩阵存在,则 $\boldsymbol{H}^{-1}$已知,可根据式(8-21)较简洁地估计出原始图像 $\boldsymbol{f}$ 的最佳估计值$\hat{\boldsymbol{f}}$,即当 $\boldsymbol{H}$ 逆作用于退化图像 $g(x,y)$时,可以实现原始图像的最小二乘方误差最小意义上的无约束估计。

8.3.2 无约束复原的奇异性

从理论上分析,由于无约束复原的处理方法仅涉及代数知识,与微积分运算无关,因此该方法简单易行,根据式(8-21)可以对退化图像 $g(x,y)$进行复原。但由于该处理方法依赖于矩阵 $\boldsymbol{H}$ 的逆矩阵,该方法存在一定的局限性。若 $\boldsymbol{H}$ 矩阵奇异,则 $\boldsymbol{H}^{-1}$不存在,这时无论是 $\boldsymbol{H}^{-1}$还是$(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}$都不存在,就无法通过式(8-21)对图像进行复原。$\boldsymbol{H}$ 逆矩阵不存在时,这种现象称为无约束复原方法的奇异性。

实际上,无约束复原的奇异性还可以进一步分析,对式(8-21)进行傅里叶变换可得

$$F(u,v) = \frac{G(u,v)}{H(u,v)} \tag{8-22}$$

由此可得,采用式(8-21)进行图像复原时,复原图像的傅里叶变换是基于退化图像的傅里叶变换 $G(u,v)$与系统函数的傅里叶变换 $H(u,v)$之比值。若 $\boldsymbol{H}^{-1}$不存在,则表明 $H(u,v)$在 u-v 平面上的个别点或某些区域其值等于 0,即 $G(u,v)/H(u,v)$出现了零点,这时,即使没有噪声,一般也无法根据 $g(x,y)$恢复原图像 $f(x,y)$。

需要指出的是,有时理论上 $G(u,v)/H(u,v)$并没有出现零点,$H(u,v)$只是接近于 0,这时也可能会导致复原模型的不稳定性。这时若系统存在噪声,则噪声项将会放大。因此,退化图像中小的噪声干扰在 $H(u,v)$取值很小值的那些频谱上将对恢复图像产生很大的影响,严重影响复原效果。

8.4 逆滤波

逆滤波复原法是无约束复原方法的一种,又称为反向滤波复原法或者去模糊。逆滤波是通过变换到频域进行滤波的,属于频域图像复原方法。逆滤波复原的基本原理是根据图像退化的原因进行反向滤波。

8.4.1 逆滤波基本原理

设原始图像为 $f(x,y)$，退化图像为 $g(x,y)$，在不考虑噪声对图像退化的影响下，则系统的退化模型为

$$g(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(\alpha,\beta)h(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{8-23}$$

根据傅里叶变换的卷积定理可得

$$G(u,v)=H(u,v)F(u,v) \tag{8-24}$$

式中，$G(u,v)$、$H(u,v)$、$F(u,v)$ 分别是退化图像 $g(x,y)$、点扩散函数 $h(x,y)$、原始图像 $f(x,y)$ 的傅里叶变换。由式(8-24)，可得

$$F(u,v)=\frac{G(u,v)}{H(u,v)} \tag{8-25}$$

$$f(x,y)=F^{-1}[F(u,v)]=F^{-1}\left[\frac{G(u,v)}{H(u,v)}\right] \tag{8-26}$$

式(8-26)表示了逆滤波法图像复原的基本原理。在存在有噪声的情况下，逆滤波原理可写为

$$G(u,v)=H(u,v)F(u,v)+N(u,v)$$

$$F(u,v)=\frac{G(u,v)}{H(u,v)}-\frac{N(u,v)}{H(u,v)} \tag{8-27}$$

式中，$N(u,v)$ 是噪声 $n(x,y)$ 的傅里叶变换。

8.4.2 逆滤波的病态条件

利用式(8-25)和式(8-27)进行复原处理时可能会发生一些奇异性，即在 u-v 平面上有些点或区域会产生 $H(u,v)=0$ 或 $H(u,v)$ 非常小的情况，在这种情况下，即使没有噪声，也无法精确地恢复 $f(x,y)$。另外，在存在噪声时，在 $H(u,v)$ 的某一邻域内，$H(u,v)$ 的值可能比 $N(u,v)$ 的值小得多，根据式(8-27)可知，噪声的影响可能会非常大，有时可能也会使 $f(x,y)$ 不能正确恢复。

为了克服这种不足，可利用有约束图像复原，也可利用噪声一般分布在高频，即噪声在高频部分衰减速度较慢，而信号的频谱随频率升高快速下降的性质，在图像复原时，只限制在频谱坐标离原点不太远的有限区域内运行，而且关心的也是信噪比高的那些频率位置。

一般来说，逆滤波方法不一定能正确地估计 $H(u,v)$ 的零点，因此必须采用一个折中的方法加以解决。实际上，逆滤波不是用 $1/H(u,v)$，而是采用另一个关于 u,v 的函数 $M(u,v)$。其基本原理如图 8-2 所示。

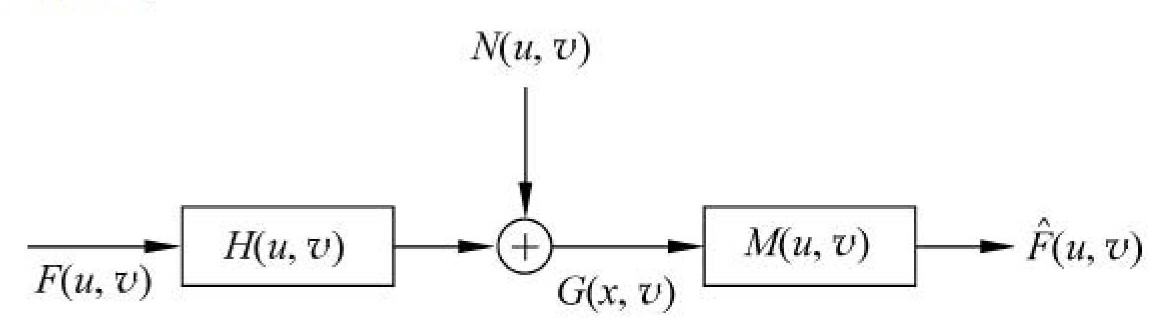


图 8-2　逆滤波系统原理框图

若系统噪声 $n(x,y)=0$，且系统 H 没有零点的情况下，设

$$M(u,v)=\frac{1}{H(u,v)} \tag{8-28}$$

实际上，图 8-2 所示的模型包括退化和恢复运算。退化和恢复总的传递函数可用 $H(u,v)M(u,v)$ 来表示，此时有

$$\hat{F}(u,v)=[H(u,v)M(u,v)]F(u,v) \tag{8-29}$$

式中，$\hat{f}(x,y)$ 是 $f(x,y)$ 的估计值；$F(u,v)$ 是 $f(x,y)$ 的傅里叶变换；$H(u,v)$ 称为输入传递函数；$M(u,v)$ 称为处理传递函数；$H(u,v)M(u,v)$ 称为输出传递函数。实际上，为了避免 $H(u,v)$ 取值太小，一种改进方法是在 $H(u,v)=0$ 的那些频谱点及其附近，可以对 $1/H(u,v)$ 的值进行设置，从而使得在这些频谱点附近 $N(u,v)/H(u,v)$ 不会对 $\hat{f}(x,y)$ 产生过大的影响。

一般情况下，$H(u,v)$ 的幅值随着离 u-v 平面原点的距离的增加而迅速下降，而噪声项 $N(u,v)$ 的幅值变化是比较平缓的。在远离 u-v 平面的原点时，$N(u,v)/H(u,v)$ 的值就会变得很大，而对于大多数图像来说 $F(u,v)$ 却变小，在这种情况下，噪声反而占优势，自然无法满意地恢复出原始图像。这一规律说明，应用逆滤波时仅在原点邻域内采用 $1/H(u,v)$ 才能奏效，换句话说，考虑到退化系统的传递函数 $H(u,v)$ 带宽比噪声的带宽要窄得多，其频率特性具有低通性质，应使 $M(u,v)$ 在下述范围内选择：

$$M(u,v)=\begin{cases}\dfrac{1}{H(u,v)} & u^2+v^2\leqslant\omega_0^2\\ 1 & u^2+v^2>\omega_0^2\end{cases} \tag{8-30}$$

ω_0 的选择应该将 $H(u,v)$ 的零点排除在此邻域之外，这种方法的不足之处是复原后的图像的振铃效果较明显。

8.4.3 逆滤波复原实例

例 8-1：逆滤波图像复原实验

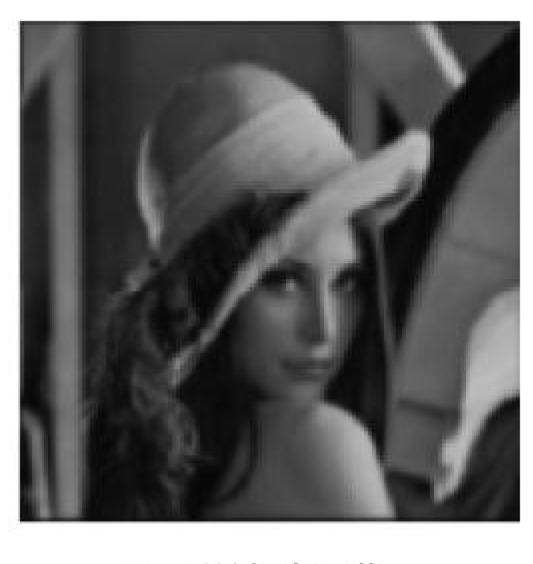
MATLAB 程序代码如下：

```
%图像复原之逆滤波程序代码
clc;
clear all;
image_o = imread('C:\Users\chen\Desktop\lena.bmp');
subplot(1,3,1);
imshow(image_o);
title('原始图像');
%在频率域进行图像退化
g = im2double(image_o);
G = fft2(g);
G = fftshift(G);
%进行图像退化
[M,N] = size(G);
[u,v] = meshgrid(1:M,1:N); %生成二维坐标系
H = exp( - 0.0025 * ((u - M/2).^2 + (v - N/2).^2).^(5/6));
G = G. * H;
```

```
% 傅里叶反变换
X = ifftshift(G);
x = ifft2(X);
image_degradation = uint8(abs(x) * 256);
subplot(1,3,2);
imshow(image_degradation);
title('退化图像');
% 逆滤波图像复原
ff = im2double(image_degradation); % 将图像灰度值归一化到 0~1 之间
% 傅里叶变换
f_Id = fft2(ff);
f_Id = fftshift(f_Id);
fH_Id = f_Id;
[M,N] = size(fH_Id);
% 逆滤波
threshold = 88;
if threshold > M/2
       fH_Id = fH_Id./(H + eps);
else
        % 在一定半径范围内进行滤波
        for i = 1:M
            for j = 1:N
                if sqrt((i - M/2).^2 + (j - N/2).^2)< threshold
                    fH_Id(i,j) = fH_Id(i,j)./(H(i,j) + eps);
                end
        end
    end
end
% 进行傅里叶逆变换
fH_Id1 = ifftshift(fH_Id);
image_restoration = ifft2(fH_Id1);
image_restoration = uint8(abs(image_restoration) * 255);
subplot(1,3,3);
imshow(image_restoration);
title('滤波半径为 88,逆滤波复原图像');
```

逆滤波图像复原效果如图 8-3 所示。


(a) 原始图像


(b) 逆滤波图像


(c) 复原图像

图 8-3 逆滤波图像复原

8.5 有约束复原法

为了在数学上更容易处理，通常在无约束复原方法的基础上附加一定的约束条件，从而在多个可能结果中选择一个最佳结果，这便是有约束的复原方法。

8.5.1 最小二乘类约束复原

无约束复原是指除了使准则函数 $J(\hat{f})=||\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}||^2$ 最小外，再没有其他约束条件。因此只需了解退化系统的传递函数或点扩散函数 H，就能利用式(8-20)或式(8-21)进行复原。但是由于传递函数 $\boldsymbol{H}$ 奇异性问题，复原只能局限在靠近原点的有限区域内进行，这就使得无约束图像复原方法具有较大的局限性。

最小二乘类约束复原是指除了要求了解关于退化系统的传递函数 $\boldsymbol{H}$ 之外，还需要知道某些噪声的统计特性或噪声与图像的某些相关情况。根据所了解的噪声的先验知识的不同，应采用不同的约束条件，可得到不同的图像复原技术。

在最小二乘类约束复原中，复原问题表现为在满足 $\|n\|^2=\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2$ 的约束条件下，要设法寻找一个最优估计 $\hat{\boldsymbol{f}}$，使得形式为 $\|\boldsymbol{Q}\hat{\boldsymbol{f}}\|^2=\|n\|^2$ 的函数最小化。求这类问题的有约束最小化，常采用拉格朗日乘数法进行处理。即寻找一个 $\hat{\boldsymbol{f}}$，使得如下准则函数最小。

$$J(\hat{\boldsymbol{f}}) = \|\boldsymbol{Q}\hat{\boldsymbol{f}}\|^2+\lambda(\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2-\|n\|^2) \tag{8-31}$$

式中，$\boldsymbol{Q}$ 为 $\hat{\boldsymbol{f}}$ 的线性算子；λ 为一常数（称为拉格朗日乘子）。

对上式求导，可得

$$\frac{\partial}{\partial \hat{\boldsymbol{f}}}J(\hat{\boldsymbol{f}}) = 2\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\hat{\boldsymbol{f}}-2\lambda\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}) = 0 \tag{8-32}$$

$$\hat{\boldsymbol{f}} = \left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+\frac{1}{\lambda}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \tag{8-33}$$

令 $\gamma=1/\lambda$

$$\hat{\boldsymbol{f}} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+\gamma\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \tag{8-34}$$

常数 γ 必须反复迭代调整，直到满足约束条件 $\|n\|^2=\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2$。求解式(8-34)的关键就是如何选用一个合适的变换矩阵 $\boldsymbol{Q}$。

相对于无约束问题，有约束条件的图像复原更符合图像退化的实际情况，因此其适应面更加广泛。对于式(8-34)，若选择不同形式的 $\boldsymbol{Q}$ 矩阵，则可得到不同类型的有约束最小二乘方类图像复原方法。如果用图像 $\boldsymbol{f}$ 和噪声的自相关矩阵 $\boldsymbol{R}_f$ 和 $\boldsymbol{R}_n$ 表示 $\boldsymbol{Q}$，就可以得到维纳滤波复原方法。如选用拉普拉斯算子形式，即使某个函数的二阶导数最小，也可推导出有约束最小平方恢复方法。

8.5.2 维纳滤波

在一般情况下，图像信号可近似为平稳随机过程，维纳滤波的基本原理是将原始图像 $\boldsymbol{f}$ 和对原始图像的估计 $\hat{\boldsymbol{f}}$ 看作随机变量，按照使 $\boldsymbol{f}$ 和估计值 $\hat{\boldsymbol{f}}$ 之间的均方误差达到最小的准则

实现图像复原。因此，维纳滤波器又称为最小均方误差滤波器，均方误差为

$$e^2 = E\{[f(x,y) - \hat{f}(x,y)]^2\}$$

式中，$E(\cdot)$表示数学期望。

设 $\boldsymbol{R}_f$ 和 $\boldsymbol{R}_n$ 是 f 和 n 的自相关矩阵，定义为

$$\boldsymbol{R}_f = E(ff^{\mathrm{T}}) \tag{8-35}$$

$$\boldsymbol{R}_n = E(nn^{\mathrm{T}}) \tag{8-36}$$

根据上述定义可知，$\boldsymbol{R}_f$ 和 $\boldsymbol{R}_n$ 均为实对称矩阵。在大多数实际图像中，相近像素点是高度相关的，而距离较远的像素点的相关性则相对较弱。通常情况下，无论 f 还是 n，其元素之间的相关不会延伸到 20～30 个像素的距离之外。因此，一般来说，自相关矩阵 $\boldsymbol{R}_f$ 和 $\boldsymbol{R}_n$ 在主对角线附近有一个非零元素带，而矩阵的右上角和左上角的区域内将接近零值。如果像素之间的相关是像素距离之函数，而不是像素位置的函数，可将 $\boldsymbol{R}_f$ 和 $\boldsymbol{R}_n$ 近似为分块循环矩阵。因而，用循环矩阵的对角化，可写成如下形式

$$\boldsymbol{R}_f = \boldsymbol{WAW}^{-1} \tag{8-37}$$

$$\boldsymbol{R}_n = \boldsymbol{WBW}^{-1} \tag{8-38}$$

式中，$\boldsymbol{W}$ 为 $MN\times MN$ 矩阵，包含 $M\times M$ 个 $N\times N$ 子矩阵。

以 $W(i,m)$表示 W 的 i 行 m 列分块矩阵，则

$$W(i,m) = \mathrm{e}^{\mathrm{j}\frac{2\pi}{M}im} W_N \quad i,m = 0,1,2,\cdots,M-1 \tag{8-39}$$

W_N是 $N\times N$ 矩阵，以 $W(k,n)$表示 k 行 n 列元素，则有

$$W_N(k,n) = \mathrm{e}^{\mathrm{j}\frac{2\pi}{M}kn} \quad k,n = 0,1,2,\cdots,M-1 \tag{8-40}$$

矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$ 的元素分别为矩阵 $\boldsymbol{R}_f$ 和 $\boldsymbol{R}_n$ 中的自相关元素的傅里叶变换，这些自相关的傅里叶变换分别定义为 $f_e(x,y)$和 $n_e(x,y)$的谱密度 $P_f(u,v)$和 $P_n(u,v)$。

维纳滤波器可以由式(8-34)导出，定义 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{R}_f^{-1}\boldsymbol{R}_n$，代入(8-34)有

$$\begin{aligned}\hat{\boldsymbol{f}} &= (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} + \gamma\boldsymbol{R}_f^{-1}\boldsymbol{R}_n)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \\ &= (\boldsymbol{WD}^*\boldsymbol{DW}^{-1} + \gamma\boldsymbol{WA}^{-1}\boldsymbol{BW}^{-1})^{-1}\boldsymbol{WD}^*\boldsymbol{W}^{-1}\boldsymbol{g}\end{aligned}$$

因此可得

$$\boldsymbol{W}^{-1}\hat{\boldsymbol{f}} = (\boldsymbol{D}^*\boldsymbol{D} + \gamma\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{D}^*\boldsymbol{W}^{-1}\boldsymbol{g} \tag{8-41}$$

若 $M=N$，则有

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma\dfrac{P_n(u,v))}{P_f(u,v)}}\right]G(u,v) \\ &= \left[\frac{1}{H(u,v)}\cdot\frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma\dfrac{P_n(u,v))}{P_f(u,v)}}\right]G(u,v) \quad u,v = 0,1,2,\cdots,N-1\end{aligned} \tag{8-42}$$

若 $\gamma=1$，则称为维纳滤波器，当无噪声影响时，由于 $P_n(u,v)=0$，则退化为逆滤波器，又称为理想的逆滤波器，因此，逆滤波器是维纳滤波器的一种特殊情况。需要注意的是，$\gamma=1$并不是在有约束条件下的最佳解，此时并不满足约束条件 $\|n\|^2=\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2$。若 γ 变参数，则称为变参数维纳滤波器。Slepian 将维纳去卷积推广用于处理包括大气扰动引起

的随机点扩散函数的情况，随后，Pratt 和 Habibl 还提出了提高维纳去卷积计算效率的方法。

维纳去卷积提供了一种在有噪声情况下导出去卷积传递函数的最优方法，但如下三个问题限制了它的有效性。

(1) 当图像复原的目的是供人观察时，均方误差(MSE)准则并不是一个特别好的优化准则。这是因为 MSE 准则不管其在图像中的位置对所有误差都赋以同样的权，而人眼则对暗处和高梯度区域的误差比其他区域的误差具有较大的容忍性。由于使均方误差最小化，维纳滤波器以一种并非最适合人眼的方式对图像进行了平滑。

(2) 经典的维纳去卷积不能处理具有空间可变点扩散函数的情形，如存在彗差、散差、表面像场弯曲以及含旋转的运动模糊等情况。

(3) 这种技术不能处理非平稳信号和噪声的一般情形。许多图像都是高度非平稳的，有着被陡峭边缘分开的大块平坦区域。此外，一些重要的噪声源具有与局部灰度有关的特性。维纳滤波图像复原效果如例 8-2 所示。

例 8-2：维纳滤波图像复原效果

图 8-4(a)为原始图像，图 8-4(b)为卷积模糊图像，图 8-4(c)是采用有约束维纳滤波器方法对图 8-4(b)所示的退化图像进行滤波，得到的复原图像。通过该实验可以看出，因为卷积模糊，图像质量明显下降，经过维纳滤波器复原以后，图像在清晰度和明亮程度上也有较大幅度提升。



(a) 原始图像



(b) 卷积模糊图像



(c) 复原图像

图 8-4　有约束图像复原

8.5.3　功率谱均衡

维纳滤波器是按照使 $f(x,y)$ 和估计值 $\hat{f}(x,y)$ 之间的均方误差达到最小的准则实现图像复原的，若改变上述准则，使原始图像 $f(x,y)$ 的功率谱 $P_f(u,v)$ 和图像估计值 $\hat{f}(x,y)$ 的功率谱 $P_{\hat{f}}(u,v)$ 相等，则称为功率谱均衡复原方法，即

$$P_f(u,v) = P_{\hat{f}}(u,v)$$

根据上述准则，Cannon 首先导出，如下形式的滤波器可使退化图像的功率谱复原至其原先的幅度，即

$$G(u,v) = \left[\frac{P_f(u,v)}{|H(u,v)|^2 + \gamma P_f(u,v) + P_n(u,v)}\right]^{\frac{1}{2}} \tag{8-43}$$

与维纳滤波器类似，式(8-43)所示的功率谱均衡滤波器也是无相移的实偶函数，它可用于无相移或相移可用其他方法确定的模糊传递函数。

功率谱均衡滤波器(8-43)与维纳滤波器是十分相似的。当无噪声时，这两处滤波器都简化为逆滤波器(即直接的去卷积)；当无信号时，这两种滤波器都完全截止。然而，不同的是功率谱均衡滤波器在模糊传递函数 $H(u,v)$为零处并不截止到零。

功率谱均衡滤波器具有相当强的图像复原能力，在某些情况下其性能优于维纳滤波器。功率谱均衡滤波器有时也称为同态滤波器。

8.5.4 几何均值滤波器

分析如下形式的复原滤波器传递函数：

$$G(u,v)=\left[\frac{H^*(u,v)}{|H(u,v)|^2}\right]^{\alpha}\left[\frac{H^*(u,v)}{|H(u,v)|^2+\gamma\frac{P_n(u,v)}{P_f(u,v)}}\right]^{(1-\alpha)} \tag{8-44}$$

式中，α 和 γ 为正的实常数。该滤波器可以认为是前面讨论过的几种滤波器的一般形式，传递函数中包含参数 α 和 γ。当 $\alpha=1$ 时，式(8-44)就成为去卷积滤波器；若 $\alpha=1/2,\gamma=1$，则式(8-44)就成为功率均衡滤波器。

进一步分析还可发现 $\alpha=1/2$ 时，式(8-44)定义的是普通去卷积和维纳去卷积的几何平均，由此可以看出，几何均值滤波器是基于逆滤波和维纳滤波器的进一步改进。因此，可以称式(8-44)所定义的滤波器为几何均值滤波器。

在式(8-44)中，若 $\alpha=0$，就可得到如下参数化的维纳滤波器：

$$G(u,v)=\frac{H^*(u,v)}{|H(u,v)|^2+\gamma\frac{P_n(u,v)}{P_f(u,v)}} \tag{8-45}$$

当 $\gamma=1$ 时，它就成了维纳滤波器；而当 $\gamma=0$ 时，它就变成了单纯的去卷积。一般来说，可通过选择 γ 的数值来获得所希望的维纳平滑效果。

式(8-44)代表一个很一般的复原滤波器，可用于具有线性与空间不变的模糊函数以及具有加性、不相关噪声的复原。在存在轻微模糊的适度噪声条件下，Andlews 和 Hunt 对式(8-44)所定义的滤波器的复原能力进行了研究，两人的研究结果表明，单纯去卷积滤波器的效果最差；而维纳滤波器会产生人眼可观察到的较严重的低通滤波效应。在同样条件下，$\gamma<1$ 的参数化维纳滤波和几何均值滤波器则可以给出更满意的效果。

8.6 匀速直线运动的模糊恢复

8.6.1 匀速运动模糊模型

在获取图像时，由于景物和摄像机之间可能会产生相对运动，往往造成图像的模糊。由于匀速直线运动是其他运动的基础，也是合成其他复杂运动的成分。因此，一般情况下，均匀直线运动所造成的模糊图像的恢复问题更具有应用价值和普遍意义。由于匀速运动退化(模糊)后的图像可表示为

$$g(x,y)=\int_0^T f(x-x_0(t),y-y_0(t))\mathrm{d}t \tag{8-46}$$

这就是成像具有相对运动时的图像退化模型，其中 $g(x,y)$ 为模糊图像，$x_0(t)$、$y_0(t)$ 分别为图像 $f(x,y)$ 在 x 和 y 方向上的变化分量，t 为运动时间。

8.6.2 匀速运动复原方法

对上述运动退化模型进行傅里叶变换可得

$$\begin{aligned} G(u,v) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left\{\int_0^T f[(x-x_0(t),y-y_0(t)]\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y\right\}\mathrm{d}t \end{aligned} \tag{8-47}$$

令

$$H(u,v) = \int_0^T \mathrm{e}^{-\mathrm{j}2\pi(ux_0(t)+vy_0(t))}\mathrm{d}t$$

根据傅里叶变换的性质，对上式进一步推导可得

$$G(u,v) = H(u,v)F(u,v) \tag{8-48}$$

上式就是运动退化模型的傅里叶变换形式，由此可知，若已知函数 $x_0(t)$ 和 $y_0(t)$ 的性质，则可以恢复原图像 $f(x,y)$。

如果模糊图像是因为景物在 x 方向的匀速直线运动而产生，则模糊后图像任意点的灰度值可表示为

$$g(x,y) = \int_0^T f[x-x_0(t),y]\mathrm{d}t \tag{8-49}$$

式中，$x_0(t)$ 是景物在 x 方向上的运动分量。

设图像在 x 方向的总位移为 a，总的运动时间为 T，可得

$$x_0(t) = \frac{at}{T} \tag{8-50}$$

若只考虑 x 方向的运动，则 $y_0(t)=0$，于是可得

$$\begin{aligned} H(u,v) &= \int_0^T \mathrm{e}^{-\mathrm{j}2\pi(ux_0(t)+vy_0(t))}\mathrm{d}t \\ &= \int_0^T \mathrm{e}^{-\mathrm{j}2\pi(ux_0(t))}\mathrm{d}t \\ &= \frac{T}{\pi au}\sin(\pi au)\mathrm{e}^{-\mathrm{j}\pi au} \end{aligned} \tag{8-51}$$

经推导可得，水平方向的匀速直线运动产生的图像模糊的数学模型为

$$g(x,y) = \int_0^T f\left[x-\frac{at}{T},y\right]\mathrm{d}t \tag{8-52}$$

其恢复近似公式为

$$f(x,y) \approx A - \mathrm{int}\left(\frac{x}{a}\right)g'\left\{\left[x-\mathrm{int}\left(\frac{x}{a}\right)a\right],y\right\} + \sum_{k=0}^{m} g'[(x-ka),y] \tag{8-53}$$

式中，g' 表示求导数运算。

同样的方法，假定 $x_0(t)=0$，可以恢复垂直方向匀速直线运动所产生的图像模糊，其图像模糊模型及近似恢复公式为

$$g(x,y) = \int_0^T f\left[x,y-\frac{bt}{T}\right]\mathrm{d}t \tag{8-54}$$

$$f(x,y) \approx A - \mathrm{int}\left(\frac{x}{a}\right)g'\left\{x,\left[y - \mathrm{int}\left(\frac{x}{a}\right)b\right]\right\} + \sum_{k=0}^{m} g'[x,(y-kb)] \tag{8-55}$$

式中，b 为图像在 y 方向的总位移。

若 x 方向和 y 方向均有运动变化，则有

$$y_0(t) = \frac{bt}{T} \tag{8-56}$$

因此系统退化函数为

$$H(u,v) = \frac{T}{\pi(au+bv)}\sin(\pi(au+bv))\mathrm{e}^{-\mathrm{j}\pi(au+bv)} \tag{8-57}$$

8.7 其他纠正技术

当图像退化系统中缺乏对退化系统的了解，无法确定图像的系统函数时，就必须采取其他方法对图像进行复原。

8.7.1 几何畸变校正

数字图像在获取过程中常常由于成像系统的几何非线性误差，导致图像产生几何失真。例如，成像图像与原景物(或物体)图像相比可能出现了比例失调，甚至扭曲等现象，这类图像退化现象一般称为图像的几何畸变。如图 8-5 所示是几种常见的几何失真情况。

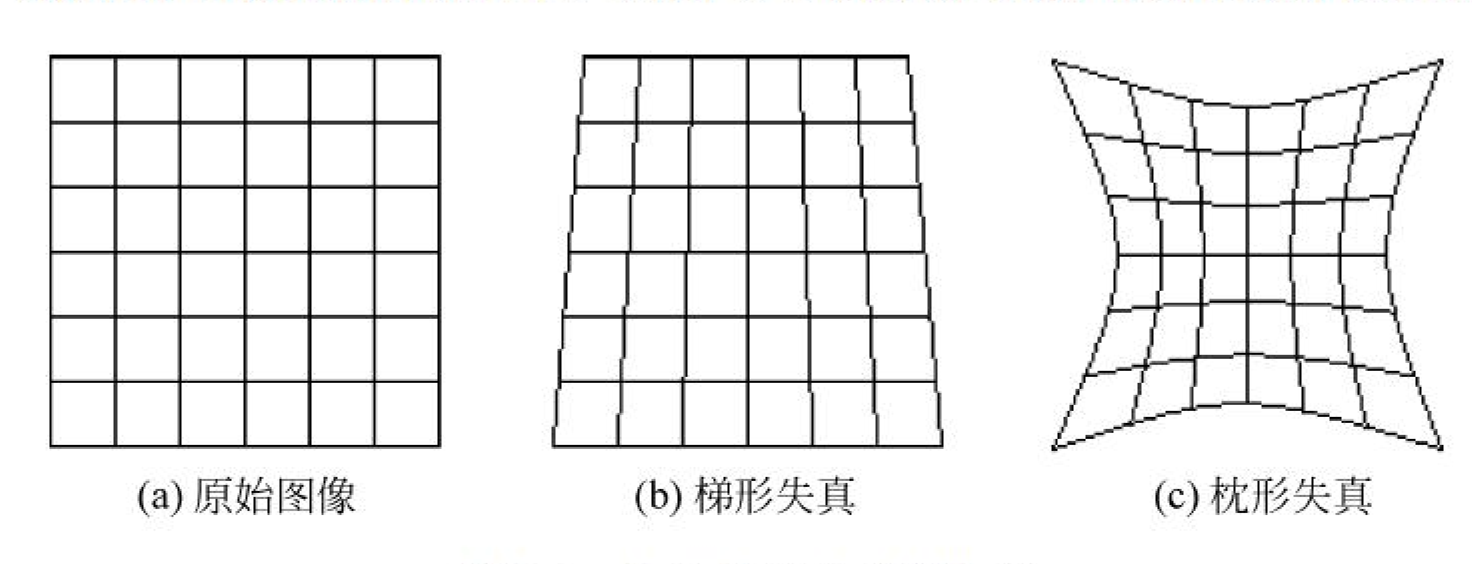


图 8-5 几种典型的几何失真

几何畸变校正的目的是需对失真的图像进行精确的几何校正，通常是先确定一幅图像为基准，然后去校正另一幅图像的几何形状。因此，几何畸变校正一般分两步进行：第一步是图像空间坐标的变换；第二步是重新确定在校正空间各像素点的取值。

1. 空间几何坐标变换

按照一幅标准图像 $g(u,v)$ 或一组基准点去校正另一幅几何失真图像 $f(x,y)$，称为空间几何坐标变换。根据几何变换理论，即根据两幅图像的一些已知对应点对或控制点对建立起映射关系，将失真图像的 x-y 坐标系变换到标准图像 u-v 坐标系，从而实现失真图像按标准图像的几何位置校正，使 $f(x,y)$ 中的每一像点都可在 $g(u,v)$ 中找到对应像点。

2. 三角形线性法

图像的几何失真虽然是非线性的，但在一个很小的局部区域内可近似认为是线性的。基于这一假设，将标准图像和被校正图像之间的控制点对划分成一系列小三角形区域，三角形顶点为三个控制点。根据几何变换理论，在三角形区内满足以下线性关系：

$$\begin{cases} x = au + bv + c \\ y = du + ev + f \end{cases} \tag{8-58}$$

若三个控制点对在两个坐标系中的位置分别为(x_1,y_1)、(x_2,y_2)、(x_3,y_3)和(u_1,v_1)、(u_2,v_2)、(u_3,v_3)，因此，应用以下校正线性方程组进行校正。

$$\begin{cases} x_1 = au_1 + bv_1 + c \\ x_2 = au_2 + bv_2 + c \\ x_3 = au_3 + bv_3 + c \\ y_1 = du_1 + ev_1 + f \\ y_2 = du_2 + ev_2 + f \\ y_3 = du_3 + ev_3 + f \end{cases} \tag{8-59}$$

通过求解上述方程组求出 a、b、c、d、e、f 六个系数，利用式(8-58)对该三角形区内其他像点进行校正。同样的方法，求解不同的三角形区域校正方程组，对不同区域进行几何校正。

三角形区域线性方程组求解法不需要了解退化系统的系统函数，通过观察失真图像，划分有特征的几何畸变三角形区域，采用计算机求解线性方程组，简单易行，能满足某些特性的几何畸变情况，并具有一定的校正精度。

需要注意的是，这是以局部小范围内的线性校正方法去处理大范围内的非线性失真，所以三角形区域的数量不能过少。一般情况下，所选择的控制点对越多，分布越均匀，三角形区域的面积越小，则校正的精度越高。但是控制点过多又会导致计算量的增加，因此需要根据退化图像的具体情况综合考虑区域的划分或控制点对的数量。

3. 灰度值的确定

理想状态下，几何畸变图像经校正以后，在校正空间中各像点的灰度值等于被校正图像对应点的灰度值。但经过校正后，图像的一些像素点因为位置的变化可能会拉伸（分散）或压缩，不一定都恰好落在坐标的整数点上，这时需要采用几何变换方法中的内插法或其他方法来获得这些像素点的灰度值。

具体插值方法的选择应根据图像的特点、灰度值分布情况等因素，恰当选择最近邻点法、双线性插值法或三次内插法等几何变换方法。

8.7.2 盲目图像复原

盲目图像复原法是指在没有图像退化先验知识、对退化系统了解不足的条件下，通过观察退化图像的多个图像并以某种方式抽出退化信息，进行图像复原的方法。在这种条件下，具有加性噪声的模糊图像复原一般有直接测量法和间接估计法两种方法。

1. 直接测量法

采用直接测量法盲目复原图像时，需要测量图像的模糊脉冲响应和噪声功率谱或协方差函数（矩阵）。在所观察的景物中，点光源能量往往直接指示出冲激响应。此外，图像边缘陡峭与否也能用来推测模糊冲激响应。一般的经验是，在背景亮度相对恒定的区域内测量图像的协方差，可以估计出观测图像的噪声协方差函数或矩阵。

2. 间接估计法

间接估计法复原图像类似于多图像平均处理法。如对一个景物连续拍摄 N 次，若每一

次获取的图像可表示为

$$g_i(x,y) = f(x,y) + n_i(x,y) \quad i = 1,2,3,\cdots,N \tag{8-60}$$

式中，$f(x,y)$表示原始图像；$g_i(x,y)$表示第 i 次获取的图像；$n_i(x,y)$表示第 i 次获取图像中的噪声。

对上述 N 次图像求平均，有

$$f(x,y) = \frac{1}{N}\sum_{i=1}^{N} g_i(x,y) - \frac{1}{N}\sum_{i=1}^{N} n_i(x,y) \tag{8-61}$$

由于一般情况下噪声 $n_i(x,y)$是白色高斯噪声，当观察次数较多时，其所有点上的数学期望为 0。由此可得

$$f(x,y) = \frac{1}{N}\sum_{i=1}^{N} g_i(x,y) \tag{8-62}$$

上式表明，通过适当次数的观察退化图像，然后进行求平均运算，可以获取真实图像。此外，盲目图像复原的间接估计法也可以利用时间上平均的概念去掉图像中的模糊。

8.8 中值滤波

中值滤波可以用于因脉冲噪声等原因而引起的退化图像的复原。中值滤波是一种非线性信号处理方法，是由 J. W. Tuky 在 1971 年提出的，它可以克服线性滤波如最小均方滤波、均值滤波给图像边缘带来的模糊，可以既去除噪声又能保护图像的边缘，从而获得较满意的图像复原效果。而且，中值滤波在实际运算过程中不需要图像的统计特性，这也带来不少方便。但对一些细节多，特别是点、线、尖顶细节较多的图像不宜采用中值滤波的方法。

8.8.1 中值滤波基本原理

中值滤波的原理非常简单，就是对一个滑动窗口(含有奇数点)内的所有像素灰度值排序，并用其中值代替窗口中心像素的原来灰度值。假设窗口内有 5 个点，其值分别为 80、90、200、110、120，那么此窗口内各点的中值即为 110。

设有一个一维序列 $x_1,x_2,\cdots,x_n$，取窗口长度为 m(m 为奇数)，对此一维序列进行中值滤波，就是从输入序列中相继抽出 m 个数 $x_{i-v},\cdots,x_i,\cdots,x_{i+v}$，其中 x_i 为窗口中心点值，$v=(m-1)/2$。然后将这 m 个点按其数值大小排序，取序列的中值作为滤波输出$\{y_i\}$，即

$$y_i = \mathrm{Med}\{x_i\} = \mathrm{Med}\{x_{i-v},\cdots,x_i,\cdots,x_{i+v}\} \quad i \in N \tag{8-63}$$

例如，若有以下输入序列：

$\{x_i\}=\{0\ 0\ 0\ 8\ 0\ 0\ 2\ 3\ 2\ 0\ 2\ 3\ 2\ 0\ 3\ 5\ 3\ 0\ 3\ 5\ 3\ 0\ 0\ 2\ 3\ 4\ 5\ 5\ 5\ 5\ 5\ 0\ 0\ 0\}$

序列中 8 是脉冲噪声，中间一段为振荡，后面是一段斜坡和跳变，如采用长度为 3 的滤波窗口，则滤波的结果为

$\{y_i\}=\{0\ 0\ 0\ 0\ 0\ 0\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 0\ 0\ 2\ 3\ 4\ 5\ 5\ 5\ 5\ 5\ 0\ 0\ 0\}$

显然，经中值滤波后，脉冲噪声 8 被滤掉了，振荡被平滑掉了，斜坡和阶跃部分被保存了下来。

如图 8-6 所示，分别应用 5 个像素的窗口($m=5$)对阶跃函数、斜坡函数、单脉冲函数、双脉冲函数、三脉冲函数、三角形函数等几种基本信号进行中值滤波和均值滤波，图中左边

一列为原始信号，中间一列为均值滤波结果，右边一列为中值滤波结果。

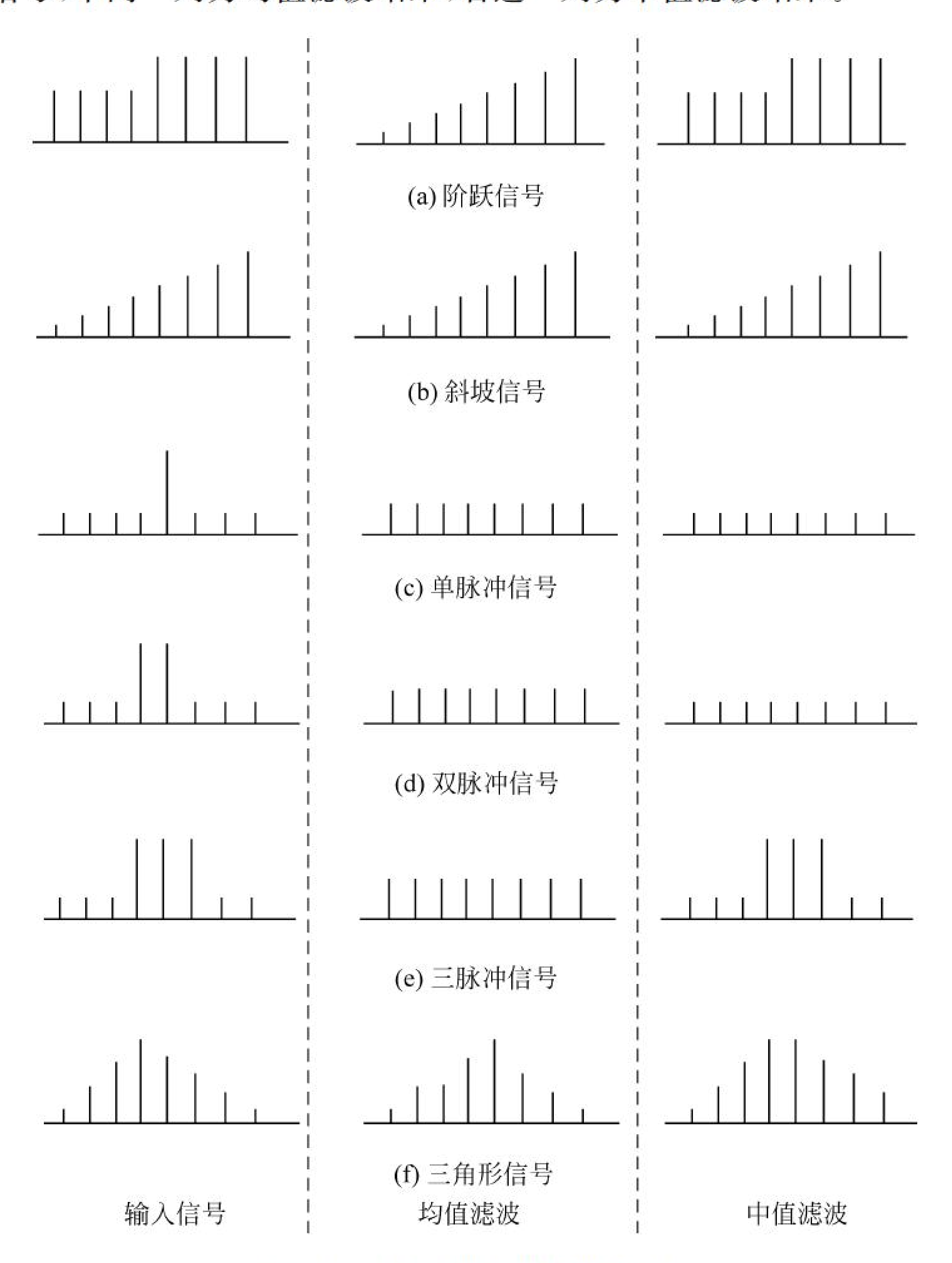


图 8-6　中值滤波和均值滤波比较

图 8-6(a)和(b)的滤波结果表明，经中值滤波器作用后，阶跃函数和斜坡函数保持不变；图 8-6(c)、(d)和(e)的滤波结果表明，经中值滤波器作用后，周期小于 $m/2$(窗口之半)的脉冲将受到抑制；而图 8-6(f)表明，经中值滤波器作用后，三角形顶部变平。

中值滤波的概念很容易推广到二维数字图像，设(x_{ij}(i,j)$\in I^2$)表示图像中各点的灰度值，滤波窗口为 A 的二维中值滤波可定义为

$$y_{ij} = \underset{A}{\mathrm{Med}}\{x_{ij}\} \tag{8-64}$$

二维中值滤波的窗口形状和尺寸设计对滤波的效果影响较大，不同的图像内容和不同的应用要求，往往采用不同的窗口形状和尺寸。如图 8-7 所示为几种二维中值滤波窗口形状，常用的有方形、圆形、十字形等。窗口大小一般由 3 到 5 逐步增大，直到获得满意的滤波效果为止。

一般情况下，对于有缓变的较长轮廓线物体的图像，采用方形或圆形为宜；对于包含有尖顶角物体的图像，适宜用十字形窗口。而窗口大小则以不超过图像中最小有效物体的尺

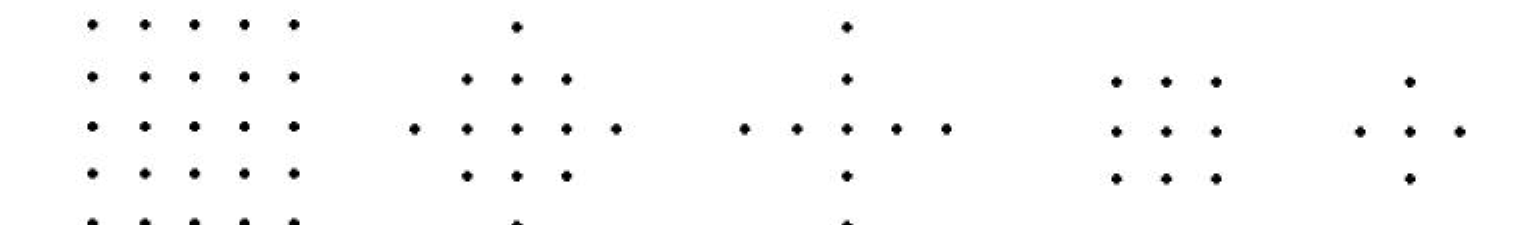


图 8-7 常用中值滤波窗口

寸为宜。使用二维中值滤波需要引起注意的是要保持图像中有效的细线状物体，如果图像中点、线、尖角细节较多，则不宜采用中值滤波。

8.8.2 中值滤波特性

1. 中值滤波对特定输入信号的不变性

对一些特定的输入信号，例如当被滤波信号在窗口内单调增加或单调减少的序列，中值滤波输出信号具有不变性。

2. 中值滤波的去噪声特性

对于零均值正态分布的噪声输入信号，中值滤波后的噪声方差 σ_{med}^2 近似为

$$\sigma_{\mathrm{med}}^2=\frac{1}{4mf^2(\bar{m})}\approx\frac{\sigma_i^2}{m+\pi/2-1}\cdot\frac{\pi}{2} \tag{8-65}$$

式中，σ_i^2 表示输入噪声功率(方差)；m 表示中值滤波窗口长度；$\bar{m}$ 表示输入噪声均值；$f(\bar{m})$ 表示输入信号噪声密度函数。

根据上式可知，中值滤波的输出与输入噪声的密度分布直接相关。同样可以得出，均值滤波的输出噪声方差 σ_{ave} 为

$$\sigma_{\mathrm{ave}}=\frac{1}{m}\sigma_i^2 \tag{8-66}$$

比较上述两式可以得出，对于随机噪声的抑制能力，中值滤波比均值滤波略差，而对于脉冲干扰，特别是脉冲宽度小于 $m/2$ 及相距较远的窄脉冲干扰，中值滤波的性能要好于均值滤波。

3. 中值滤波的频谱特性

一般情况下，信号经中值滤波之后，其均值较为平坦，频谱特性起伏不大。因此，可以近似认为，经中值滤波之后，信号频谱近似保持不变。

8.8.3 加权中值滤波

上述中值滤波方法，窗口内各点对输出结果的作用是相同的，如果希望强调中间点或中间几个点(或距中间最近的若干点)的作用，可以采用加权中值滤波方法。如果适当地选取窗口内各点的权重，加权中值滤波比简单中值滤波能更好地从被噪声污染的图像中恢复出阶跃边缘以及图像的其他细节。

加权中值滤波的基本原理是改变窗口中变量的个数，可以使一个以上的变量等于同一点的值，然后对扩张后的数字集求中值。以窗口为 3 的一维加权中值滤波为例，可表示为

$$\begin{aligned}y_i&=\mathrm{Weight_Med}\{x_i\}=\mathrm{Weight_Med}\{x_{i-1},x_i,x_{i+1}\}\\&=\mathrm{Med}\{x_{i-1},x_{i-1},x_i,x_i,x_i,x_{i+1},x_{i+1}\}\quad i\in N\end{aligned} \tag{8-67}$$

在窗口内，中间点取奇数，两边点取对称数，即位于窗口中间的像素重复两次，位于窗口

边缘的两个像素各重复一次，组成新的序列，然后对新的序列再按常规中值滤波方法进行处理。

习题

1. 试简述导致图像质量退化的原因通常有哪些。
2. 图像复原与图像增强有哪些主要区别?
3. 试简述图像复原的基本原理。
4. 什么是图像退化模型的冲激响应函数?
5. 试写出一维离散退化模型的矩阵形式。
6. 降低图像退化模型复原处理的运算量一般有哪些主要方法?
7. 什么是无约束复原? 其基本原理是什么?
8. 无约束复原的奇异性对图像复原有哪些影响?
9. 简述如何克服图像复原过程中的奇异性影响。
10. 什么是有约束复原? 一般在什么条件下选择有约束复原?
11. 什么是逆滤波图像复原法? 它的基本原理是什么?
12. 选择一幅图像，根据逆滤波原理编写 MATLAB 程序，对图像进行逆滤波复原，观察不同滤波半径的复原效果。
13. 试简述有哪些因素会影响维纳滤波的有效性?
14. 编写 MATLAB 程序实现对图像的维纳滤波，分析维纳滤波效果。
15. 试对图 8-8 进行 3×3 的中值滤波处理，并分析处理结果。

1	1	7	1	8	1	7	1
1	1	1	1	5	1	1	1
1	1	1	5	5	5	1	1
7	1	1	5	5	5	1	1
1	1	1	5	5	5	1	8
1	8	1	1	5	1	1	1
1	8	1	1	5	1	1	8
1	1	1	1	5	1	1	1
1	1	7	1	8	1	7	1

图 8-8 对图形进行中值滤波处理

16. 根据中值滤波的基本原理，分析中值滤波的主要功能有哪些。
17. 什么是图像的几何畸变? 如何消除图像的几何畸变?
18. 什么是盲目图像复原? 盲目图像复原有哪些方法?

第 9 章 图像编码

CHAPTER 9

图像编码也称图像压缩，是指在保证一定质量（信噪比或主观评价）的条件下，按照一定的规则、以较少比特数表示源图像的一种方法。信息科学的奠基人香农早已论证，不论语音信号还是图像信号，信号中都包含很多冗余信息。图像中常见的各种冗余信息主要包括空间冗余、时间冗余、信息熵冗余、结构冗余、知识冗余等，这就使得通过编码来进行压缩成为了可能。

数字图像相对于模拟图像，具有很多优点，它不仅能最大限度地避免各种噪声和干扰，而且便于利用计算机进行处理和进行信息传输与交换，可以方便地对图像进行压缩、增强、恢复、特征提取和识别等一系列处理。图像信息存在的最大问题是数据量非常庞大，庞大的数据量给计算机存储系统、处理系统、数据传输系统都提出了新的挑战。一幅分辨率为 1024×768 的 RGB 彩色图像，各单色像素需要采用 8 位表示，它的数据量为 1024×768×8×3bit，数据量超过 2.25MB。因此，如何快速有效地对图像进行编码及图像数据的存储和传输，成为了信息技术的一个研究热点和重点。

数字图像庞大的数据量给计算机存储系统、处理系统、数据传输系统都提出了新的挑战，因此，必须根据图像数据的特点对其进行压缩编码。实际上，数字图像信息具有很大的压缩潜力，因为数字图像中像素与像素之间无论在行还是列方向上的数据都存在较大的相关性，并考虑到人类视觉特性对彩色的敏感存在一定的局限性，所以完全可以将数字图像中的这部分冗余信息去掉，即允许在一定限度失真的前提下，按照某种方法对图像数据进行压缩。本章将围绕图像编码的基本原理，介绍图像数据的霍夫曼编码、香农—范诺编码、行程长度编码、算术编码、位平面编码、预测编码、变换编码、JPEG 编码标准及 MPEG 编码标准等多种编码方法。

9.1 图像编码的基本原理

编码是信息科学的经典研究课题，应用计算机进行图像处理必须首先进行图像数字化，而图像数字化的关键是编码。50 年来，图像编码技术得到了快速的发展，已经有多种成熟的编码方法得到了广泛应用，特别是所谓的第一代编码更是如此。而且，随着多媒体技术的发展，已有若干编码标准由 ITU-T 制定出来，如 JPEG、H.261、H.263、MPEG-1、MPEG-2、MPEG-4、MPEG-7、JBIG 等。数据压缩编码方法经过几十年的发展，针对不同的原始数据的特点如文本、声音、图像、视频、动画已研究出了不同的编码方法。

9.1.1 图像中的信息冗余

虽然数字图像需要占用大量的存储单元，但图像数据是高度相关的，即数字图像信息中存在大量的冗余信息。若能去掉这些冗余信息，则既可以实现对图像数据的有效压缩，同时又不会对图像的有用信息造成太大的损失。数字图像的信息冗余主要包括视觉冗余、空间冗余、时间冗余、信息熵冗余、结构冗余和知识冗余等形式。

1. 视觉冗余

视觉冗余是指人类视觉系统不敏感或不能感知的那部分图像信息。人的眼睛对图像细节和颜色的辨认能力是有限的，研究表明，人类的视觉能力最多可辨认出上千种颜色，而彩色图像一般每个像素用 24 位表示，可表示出 2^{24} 种颜色，由此可见彩色图像对人的视觉特性而言，存在大量的信息冗余，称为视觉冗余。

2. 空间冗余

图像内部相邻像素之间存在较强的相关性所产生的图像信息冗余称为时间冗余。图像中的大部分物体的表面颜色都是均匀的、连续的，图像内部相邻像素之间存在较强的相关性。因此图像数字化为像素点的 $M \times N$ 矩阵以后，矩阵中的大量相邻数据是十分接近或完全一样，这就是图像信息的空间冗余。

3. 时间冗余

对于以电视、电影等为代表的视频图像，视频图像信号播放过程中图像序列的不同帧之间存在大量的相关性，由此产生的信息冗余称为时间冗余。

4. 信息熵冗余

信息熵冗余又称为编码冗余，如果图像中平均每个像素使用的比特数大于该图像的信息熵 H，则图像中存在信息冗余，这类冗余称为信息熵冗余。

5. 结构冗余

在许多图像中，图像中可能存在很强的纹理结构或自相似形，或图像全局或不同部分之间存在很强的纹理结构或自相似性，由此产生的图像信息冗余称为结构冗余。

6. 知识冗余

在某些图像中还包含与某些先验知识有关的信息，由此产生的图像信息冗余称为知识冗余。

图像数据中存在的冗余信息为图像压缩编码提供了依据。图像编码的目的就是尽可能去除图像数据中存在的各种冗余信息，特别是空间冗余、视觉冗余及时间冗余，以尽可能少的比特编码表示一幅图像数据。例如，由于人眼的视觉特性对蓝光不敏感，因此，彩色图像编码时，就可以用较低的精度对蓝色分量进行编码。利用各种冗余信息，压缩编码技术能够很好地解决将模拟信号转换为数字信号后所产生的带宽需求增加的问题，它是使数字信号走上实用化的关键技术之一。表 9-1 列出了几种常见应用的码率。

表 9-1 常用电视信号的码率

应用种类	bit/像素	像素数/行	行数/帧	帧数/s	亮色比	压缩前	压缩后
电视电话	8	128	112	30	4∶1∶1	5.2Mb/s	56Kb/s
会议电视	8	352	288	30	4∶1∶1	36.5Mb/s	1.5～2.0Mb/s
普通电视	8	720	480	30	4∶1∶1	167Mb/s	4.0～9.0Mb/s
高清电视	8	1920	1080	30	4∶1∶1	1.18Gb/s	20.0～25.0Mb/s

9.1.2 图像编码分类

图像编码是指按照一定的格式存储图像数据的过程,而编码技术则是研究如何在满足一定的图像保真条件下,压缩表示原始图像数据的编码方法。图像编码技术不仅仅是应用线性脉冲编码调制法(线性 PCM),更主要的是利用图像信号的统计特性以及视觉对图像的生理学和心理学特性对图像进行信源编码。目前有很多流行的图像格式标准,如 BMP、PCX、TIFF、GIF、JPEG 等,它们都是采用了不同的编码方法。

在信息论中,一个典型的数字通信模型如图 9-1 所示。其中,信道编码的主要任务是解决信息传输的可靠性问题,使信号在传输过程中尽量不出错或少出错,以及产生错误之后的错误纠正。信源编码的主要任务是解决信息表示的有效性问题,即尽量提高编码效率,以最少的数码表示尽可能多的信息量,对信源实现压缩处理。图像编码的目的是要研究压缩数码率,即高效编码问题。

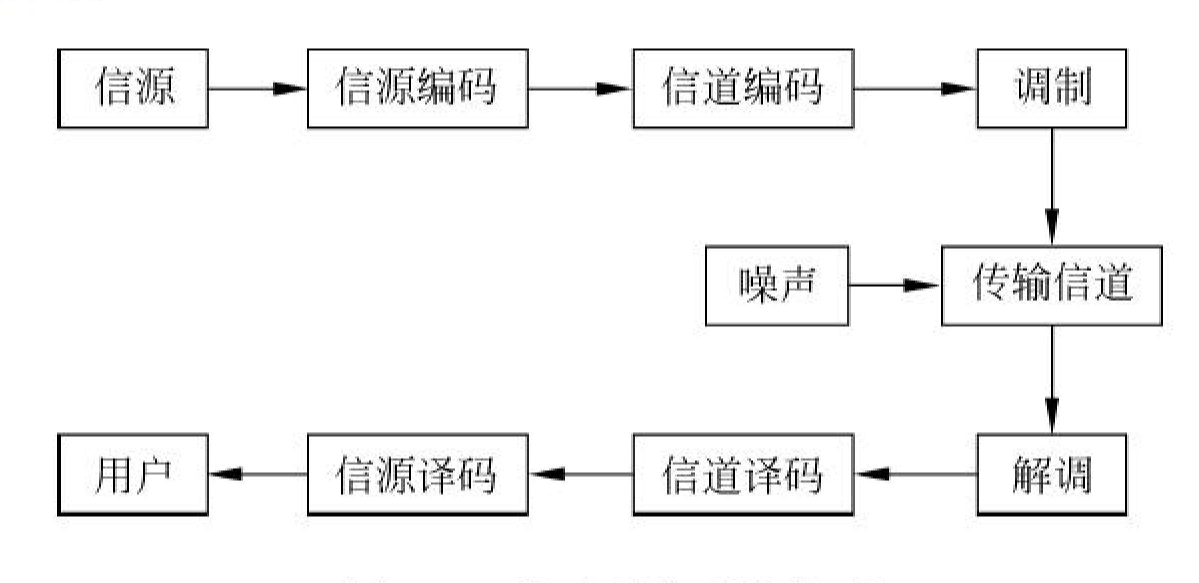


图 9-1 典型通信系统模型

图像编码属于信源编码的范畴,从不同的角度看问题就会有不同的分类方法,并没有统一的标准。根据图像的信号形式可分为模拟图像编码和数字图像编码;按图像光谱特征可分为单色图像、彩色图像和多光谱图像编码;从信号处理维数出发可以分为行内编码、帧内编码和帧间编码;从灰度概念出发可以分为二值图像和多灰度图像编码。此外,Kunt 将以消除冗余为基础的编码方法称为第一代编码,包括 PCM、DPCM、亚取样编码法,基于变换域的 DFT、DCT、Walsh-Hadamard 等编码方法以及混合编码法均属于经典的第一代编码法。而第二代编码方法主要指 Fractal 编码、金字塔编码法、小波变换编码法、基于神经元网络的编码方法、模型基编码法等新的编码方法。由此可以看出,不仅图像编码的方案是多种多样的,编码的分类也不是绝对的。

根据压缩原理进行划分,可以分为四大类,如表 9-2 所示,这些方法既适用于静止图像编码,也适用于视频图像信号编码。

表 9-2 图像编码方法分类

算法分类	编码方法举例	主要特点
熵编码	霍夫曼编码	熵编码又称为统计编码,建立在图像统计特征基础之上的数据压缩方法
	算术编码	
	RLE 编码	

续表

算法分类	编码方法举例	主要特点
预测编码	DPCM 编码	依据某一模型,根据以往的样本值对于新样本值进行预测编码
	运动补偿法	
变换编码	DCT 变换编码	将空域中描述的图像数据经过变换,转换到变换域中进行描述,达到改变能量分布的目的,实现对信源图像数据的有效压缩
	DFT 变换编码	
	小波变换编码	
混合编码	JPEG 编码	主要包括静止图像和运动图像的压缩标准系列
	MPEG 编码	

根据解码后的图像数据与原始图像数据是否完全一致,可分为有损编码与无损编码。有损编码是指对图像进行解压后重新构造的图像与原始图像存在着一定的误差。有损压缩利用了图像信息本身包含的许多冗余信息,如视觉冗余和空间冗余。由于有损压缩一般情况下可获得较好的压缩比,因此在对图像的质量要求不苛刻的情况下一般应选择有损压缩。

无损编码是指对图像数据进行解压后重新构造出来的图像与原始图像之间完全相同。行程长度编码就是无损编码的一个实例,其编码原理是在给定的数据中寻找连续重复的数值,然后用两个数值(重复数值的个数,重复数值本身)代替这些连续的数值,以达到数据压缩的目的。运用此方法处理拥有大面积色调一致的图像时,可达到很好的数据压缩效果。

需要指出的是,为了实现更高的编码效率,各种具体编码方法并不一定是孤立、单一地使用,往往需要各种方法的交叉使用。

9.1.3 图像编码新方法

图像编码新方法主要是指 Kunt 提出的第二代编码方法,包括分形编码(fractal coding)、小波变换编码(wavelet coding)、模型压缩编码(model based coding)、人工神经网络(artificial neural network)编码和基于对象的压缩编码(object based coding)等。

1)分形编码

分形编码是基于美籍波兰数学家 B. B. Mandelbrot 建立的分形几何理论而发展起来的一种新的编码方法,它主要适用于自相似性较高的图像。分形编码最大限度地利用了图像在空间域上的自相似性,即图像局部与整体之间存在的某种相似性,通过消除图像的几何冗余来压缩数据。M. Barnsley 将迭代函数系统(IFS)用于描述图像的自相似性,并用于图像编码,对某些特定图像获得了高达 10 000∶1 的压缩比。相比较而言,分形编码过程非常复杂,但解码过程较简单。因此,该方法通常用于只需对图像进行一次编码,但需多次译码的信息传播系统中。

2)小波变换编码

小波变换编码于 1989 年由 S. G. Mallat 首次使用,由于小波变换在时频两域同时具有良好的特性,因此经过小波变换后的图像具有很好的空间方向选择性,而且保持了小波的多分辨率特性,能够保持原图像在各种分辨率下的精细结构,与人的视觉特性非常吻合。

3)模型压缩编码

模型压缩编码的基本原理是在编码、解码两端分别建立相同的模型,编码时利用先验模型抽取图像中的主要特征信息并以模型参数的形式表示,解码时则根据所接收的模型参数

重建图像。模型压缩编码又称为模型编码,它是近几年发展起来的一种很有前途的低比特率编码新方法。

新编码方法一般具有以下特点:

(1) 充分考虑人的视觉特性。

(2) 合理地考虑对图像信号的分解与表述。

(3) 采用图像的合成与识别方法提高数据压缩率。

9.2　图像编码评价

50年来,图像编码的新方法不断涌现,随着众多图像压缩算法的出现以及图像处理技术在各行各业的广泛应用,图像压缩算法的评价开始受到广泛关注。一般情况下,图像压缩算法的评价主要包括编码图像的质量、图像编码效率、复杂度与适用范围等方面。

9.2.1　编码图像的质量

图像质量评价可分为客观评价和主观评价。客观评价指标主要是均方误差(MSE)和峰值信噪比(PSNR),其定义分别为

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[f(i,j) - \hat{f}(i,j)]^2 \tag{9-1}$$

$$\mathrm{PSNR} = 10\lg\left(\frac{255 \times 255}{\mathrm{MSE}}\right) \tag{9-2}$$

客观评价特点是指标具有客观性,可快速有效地评价编码图像的质量,但符合客观评价指标的图像不一定具有较好的主观质量。图像质量的好坏,既与图像本身的客观质量有关,也与人的视觉系统的感受特性有关。有时客观保真度指标完全相同的两幅图像可能会有完全不相同的视觉效果,因此,对于图像质量的评价还需采用主观保真度准则。主观评价是指由一批观察者对编码图像进行评价计分,然后依据全部评价人员的评判信息,给出图像的质量评价。主观评价的特点是与人的视觉效果相匹配,但其评判过程缓慢费时。

9.2.2　效率指标

图像编码算法的效率主要表现为平均码字长度(R)、图像的压缩比(r)、每秒传输比特数(bps)、图像编码效率(η),这些表现形式很容易相互转换。

1. 图像的信息熵

设一幅灰度级为K的图像,图像中第k级灰度出现的概率为p_k,图像大小为$M \times N$,每个像素用d比特表示,每两帧图像间隔Δt,则按信息论中信息熵的定义,数字图像的熵H由下式定义

$$H = -\sum_{k=1}^{K} p_k \log_2 p_k \tag{9-3}$$

根据上述定义可知,信息熵H表示各灰度级比特数的统计平均值。

2. 图像的平均码字长度

图像的平均码字长度R为

$$R = \sum_{k=1}^{K} B_k p_k \tag{9-4}$$

式中，B_k 表示某种图像编码方法的第 k 级灰度的码字长度。

3. 图像的编码效率

编码效率 η 定义为

$$\eta = \frac{H}{R} \times 100\% \tag{9-5}$$

4. 图像的信息冗余度

信息冗余度为

$$\upsilon = 1 - \eta \tag{9-6}$$

5. 每秒钟所需的传输比特数

每秒钟所需的传输比特数 bps 为

$$\text{bps} = \frac{M \times N \times R}{\Delta t} \tag{9-7}$$

6. 压缩比

压缩比 r 为

$$r = \frac{d}{R} \tag{9-8}$$

一般情况下，同一压缩算法对不同图像的编码效率并不一定相同。

9.2.3 复杂度与适用范围

图像编码算法的复杂度指完成图像压缩和解压所需的运算量以及实现该算法的硬件复杂度。一个好的压缩算法一般应具有高压缩比、算法简单、压缩和解缩过程快、易于硬件实现、解压后图像质量好等特点。

几乎所有的图像编码算法都具有自身的适用范围，并非对所有图像都具有高效和高质量特性。一般来说，大多数基于图像信息统计特性的压缩算法具有较广的适用范围，而一些特定的编码算法的适用范围较窄，如分形编码主要用于自相似性高的图像。

9.3 图像统计编码

统计编码又称为熵编码，它是建立在图像统计特征基础之上的数据压缩方法，根据香农的观点，信息冗余来自信源数据本身的相关性和信源内事件的概率分布不均匀性，只要找到去除相关性和改变概率分布不均匀的方法，也就找到了信源数据的统计编码方法。

9.3.1 变长最佳编码定理

在变长编码中，对出现概率大的符号编以短码字，对出现概率小的符号编以长码字。如果码字长度严格按照对应符号出现的概率大小逆序排列，则其平均码字长度为最小，这就是变长最佳编码定理。这是一种最佳编码定理，变长最佳编码定理是霍夫曼编码的理论基础。

根据信源编码理论，当平均码字长度 R 大于或等于图像熵 H 时，总可设计出一种无失真编码方法。当平均码字长度远大于图像熵时，表明该编码方法效率很低；当平均码字长

度等于或按大于方向很接近于图像熵时，称此编码方法为最佳编码，此时不会引起图像失真；当平均码字长度小于图像熵 H 时，压缩比较高，但会引起图像失真。

设变长编码所用的码元进制为 D，第 i 个符号出现的概率为 p_i，与其对应的码子长度为 N_i，则变长最佳编码的平均码字长度 R 的范围为

$$\frac{H}{\log D}+1 \geqslant R \geqslant \frac{H}{\log D} \tag{9-9}$$

而对于出现概率为 p_i 的信源符号，其码字长度存在如下关系：

$$-\log_D p_i + 1 \geqslant N_i > -\log_D p_i \tag{9-10}$$

因此，对于常用的二进制则有

$$-\log_2 p_i + 1 \geqslant N_i > -\log_2 p_i$$

由此可见，信源符号的码字长度是由信源符号自身出现的概率决定的。

9.3.2　霍夫曼编码

霍夫曼(Huffman)编码是霍夫曼于 1952 年提出的一种无损编码方法，该方法完全依据信源字符出现的概率来构造其码字，对出现概率大的字符使用较短的码字，而对出现概率低的字符使用较长的码字，从而达到压缩数据的目的。霍夫曼编码有时又称为最佳编码(一般直接叫霍夫曼编码)，最初主要用于文本文件压缩。霍夫曼编码是一种变长编码(VLC)，同时也是一种无失真编码，属于熵编码范畴。在具有相同信源概率分布的前提下，它的平均码字长度比其他任何一种编码方法都短。

1. 霍夫曼编码算法

霍夫曼编码是以信源字符的概率分布为基础的，若理论上并不知道信源字符的概率分布，可以根据对大量数据进行统计所得到的统计分布来近似代替。以统计数据代替实际概率分布，可能会导致实际应用时霍夫曼编码无法达到最佳编码效能，应用中可以根据输入数据序列自适应地匹配信源概率分布的方法，这样就能一定程度上改进霍夫曼编码的性能。

实现霍夫曼编码的基本步骤如下：

(1) 统计信源字符序列中各符号出现的概率，将各字符出现的概率按由大到小的顺序排列。

(2) 将最小的两个概率相加，并合并成新的概率，与其他概率重新按由大到小的顺序排列。

(3) 重新排列后将两个最小概率相加合并为新的概率，即重复步骤(2)直到最后两个概率之和为 1.0。

(4) 对每个概率相加的组合中，概率大的指定为 0，概率小的指定为 1(或者概率大的指定为 1，小的指定为 0)，若两者相等，则可以指定任意一个为 0，而另一个为 1。

(5) 找出由每一个信源字符到达概率为 1.0 处的路径，顺序记录沿路径的每一个 1 和 0 的数字编码。

(6) 反向写出编码，即为该信源字符的霍夫曼编码。

2. 霍夫曼编码实例

下面通过一个例子来详细说明霍夫曼编码的具体过程。

例 9-1：霍夫曼编码实例

设一幅灰度级为8的图像中，各灰度级分别用S_0、S_1、S_2、S_3、S_4、S_5、S_6、S_7表示，对应的概率分别为0.40、0.18、0.10、0.10、0.07、0.06、0.05、0.04。现对其进行霍夫曼编码。

编码过程如图9-2所示，具体步骤如下：

(1) 对信源概率由大到小排序，排列为0.40、0.18、0.10、0.10、0.07、0.06、0.05、0.04。

(2) 选出两个最小的概率0.04和0.05，相加得0.09，与其他概率重新排序0.40、0.18、0.10、0.10、0.09、0.07、0.06。

(3) 对重新排序的概率选出两个最小的概率0.06和0.07，相加得0.13，组成新的概率序列0.40、0.18、0.10、0.10、0.09、0.13。

(4) 对重新排序的概率选出两个最小的概率0.09和0.10，相加得0.19，组成新的概率序列0.40、0.18、0.13、0.10、0.19。

(5) 对重新排序的概率选出两个最小的概率0.10和0.13，相加得0.23，组成新的概率序列0.40、0.19、0.18、0.23。

(6) 对重新排序的概率选出最小的两个概率0.18和0.19，相加得0.37，组成新的概率序列0.40、0.23、0.37。

(7) 对重新排序的概率选出最小的两个概率0.23和0.37，相加得0.60，组成新的概率序列0.40、0.60。

(8) 已形成只有最后两个概率0.40和0.60，相加之和为1.0。

(9) 写出路径编码。根据以上步骤可以得出如图9-2所示编码图形。按编码一般步骤，从第一个符号开始或者从最后一个符号开始进行路径的0或1数字编码。若从最后一个符号开始，首先给最后相加的两个概率0.40和0.60分配码字，由于0.60大于0.40，因此概率0.60的路径为0，而0.40的路径1。依次类推，给每次相加的两个概率的路径分配码字0或1，最后得到如图9-2所示的编码。

(10) 形成霍夫曼编码。以对应概率为0.18的字符S_1为例，在从0.18到1.0的路径上，S_1沿路径编码依次为1、0、0，反向写出编码即为001，因此，字符S_1的霍夫曼编码为001。

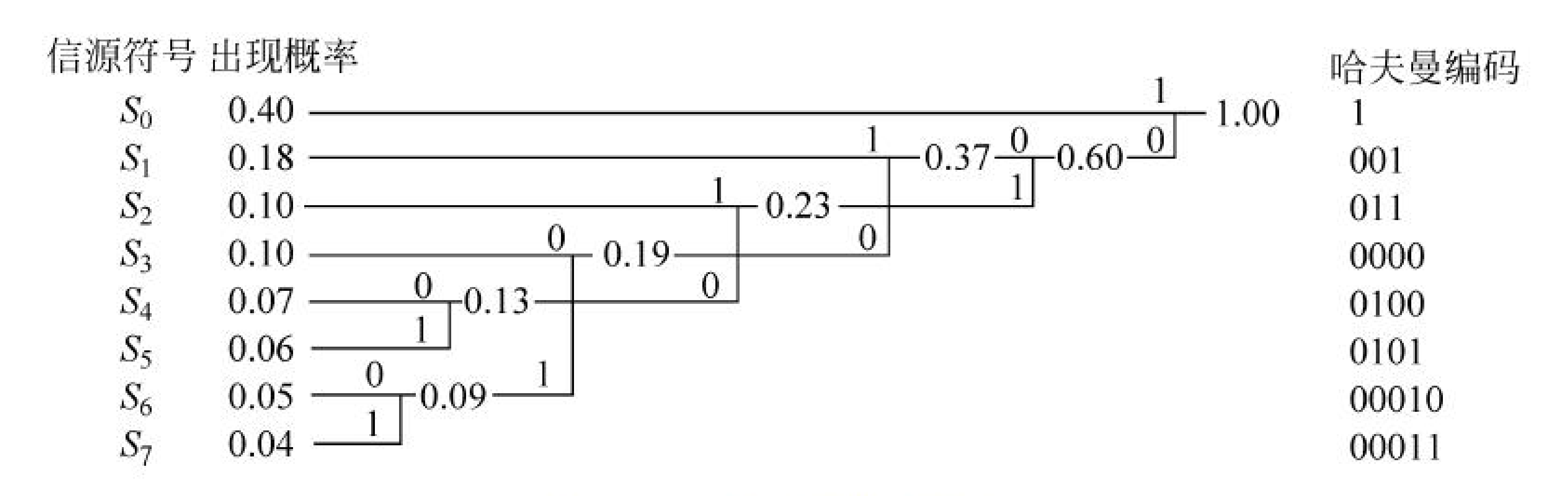


图9-2 霍夫曼编码过程

上述霍夫曼编码方法形成的码字是可识别的，即能够保证一个符号的码字不会与另一个符号的码字的前几位相同。

3. 霍夫曼编码的效率

(1) 图像信息熵为

$$H=-\sum_{k=0}^{N} p_k \log_2 p_k$$

$= -(0.40 \times \log_2 0.40 + 0.18 \times \log_2 0.18 + 2 \times 0.10\log_2 0.10 + 0.07 \times \log_2 0.07 + 0.06 \times \log_2 0.06 + 0.05 \times \log_2 0.05 + 0.04 \times \log_2 0.04$
$= 2.55$

(2) 平均码字长度为

$$R = \sum_{k=1}^{N} B_k p_k$$
$$= 1 \times 0.40 + 3 \times 0.18 + 3 \times 0.10 + 4 \times 0.10 + 4 \times 0.07 + 4 \times 0.06 + 5 \times 0.05 + 5 \times 0.04 = 2.61$$

(3) 效率为

$$\eta = \frac{H}{R} \times 100\% = \frac{2.55}{2.61} = 97.8\%$$

(4) 信息冗余度为

$$\upsilon = 1 - \eta = 2.2\%$$

根据例 9-1 的计算结果，可得如表 9-3 所示的霍夫曼编码效能表。

表 9-3　霍夫曼编码结果

信源符号	出现概率	霍夫曼码字	码字长度
S_0	0.40	1	1
S_1	0.18	001	3
S_2	0.10	011	3
S_3	0.10	0000	4
S_4	0.07	0010	4
S_5	0.06	0101	4
S_6	0.05	00010	5
S_7	0.04	00011	5
编码效能	信息熵	$H=2.55$	
	平均码长	$R=2.61$	
	编码效率	$\eta=97.8\%$	

若 8 个灰度级按如表 9-4 所示第 2 列的概率出现，即分别以 2 的负幂次方且非均匀分布的概率出现，则其霍夫曼编码效果如表 9-4 所示。

表 9-4　非均匀概率分布霍夫曼编码效能

信源符号	出现概率	霍夫曼码字	码字长度
S_0	2^{-1}	1	1
S_1	2^{-2}	01	2
S_2	2^{-3}	001	3
S_3	2^{-4}	0001	4
S_4	2^{-5}	00001	5
S_5	2^{-6}	000001	6
S_6	2^{-7}	0000001	7
S_7	2^{-7}	0000000	7

续表

信源符号	出现概率	霍夫曼码字	码字长度
编码效能	信息熵	$H=1.984\ 375$	
	平均码长	$R=1.984\ 375$	
	编码效率	$\eta=100\%$	

4. 霍夫曼编码的特点

霍夫曼编码具有以下特点：

(1) 当图像灰度值分布很不均匀时，霍夫曼编码效率比较高。以例 9-1 为例，编码效率为 97.8%，其冗余度只有 2.2%，若采用等长编码，由于灰度级为 8，根据信息论每种灰度级别至少需要 3bit 来表示，则平均码长为 3，编码效率为 85%。当信源概率是 2 的负幂次方时，编码效率为 100%。而在图像灰度值的概率分布比较均匀时，霍夫曼编码的效率就低。

(2) 对不同概率分布的信源，霍夫曼编码的编码效率各不相同。对于二进制编码体系，当信源概率为 2 的负幂次方时，霍夫曼编码的编码效率可达 100%，其平均码字长度也很短，而当信源概率为均匀分布时，其编码效果明显降低。即便是如表 9-5 所示，服从 2 的负幂次方的均匀分布信息源，由于字符概率均匀分布，其熵最大，平均编码长度很大(编码过程如图 9-3 所示)，因此从压缩比 r 等其他指标看，其编码效率较低。也就是说，在信源概率接近于均匀分布时，一般不使用霍夫曼编码。

(3) 霍夫曼编码方法产生的编码不是唯一的。编码过程中在给两个最小概率的灰度值进行编码时，既可以是大概率为 0，小概率为 1，也可以是大概率为 1，小概率为 0。另外，当两个灰度值出现的概率相等时，0 和 1 的分配也是随机的，因此编码不具唯一性。

(4) 霍夫曼编码需建立在已知图像数据信息的概率分布特性的基础上，才可以实现图像数据编码。若信源符号很多，那么码表就会很大，这必将影响到存储、编码与传输。因此，有些观点认为霍夫曼编码缺乏构造性。

(5) 霍夫曼编码方法对图像数据进行编码时，需两次读取图像数据。第一次是为了计算每个数据出现的概率，并对各数据出现的概率进行排序，在排序过程中获得各数据的编码值。第二次读取数据是以转换表格中的编码值代替图像数据存入图像编码文件中。

表 9-5 均匀概率分布霍夫曼编码效能

信源符号	出现概率	霍夫曼码字	码字长度
S_0	2^{-3}	111	3
S_1	2^{-3}	110	3
S_2	2^{-3}	101	3
S_3	2^{-3}	100	3
S_4	2^{-3}	011	3
S_5	2^{-3}	010	3
S_6	2^{-3}	001	3
S_7	2^{-3}	000	3
编码效能	信息熵	$H=3$	
	平均码长	$R=3$	
	编码效率	$\eta=100\%$	

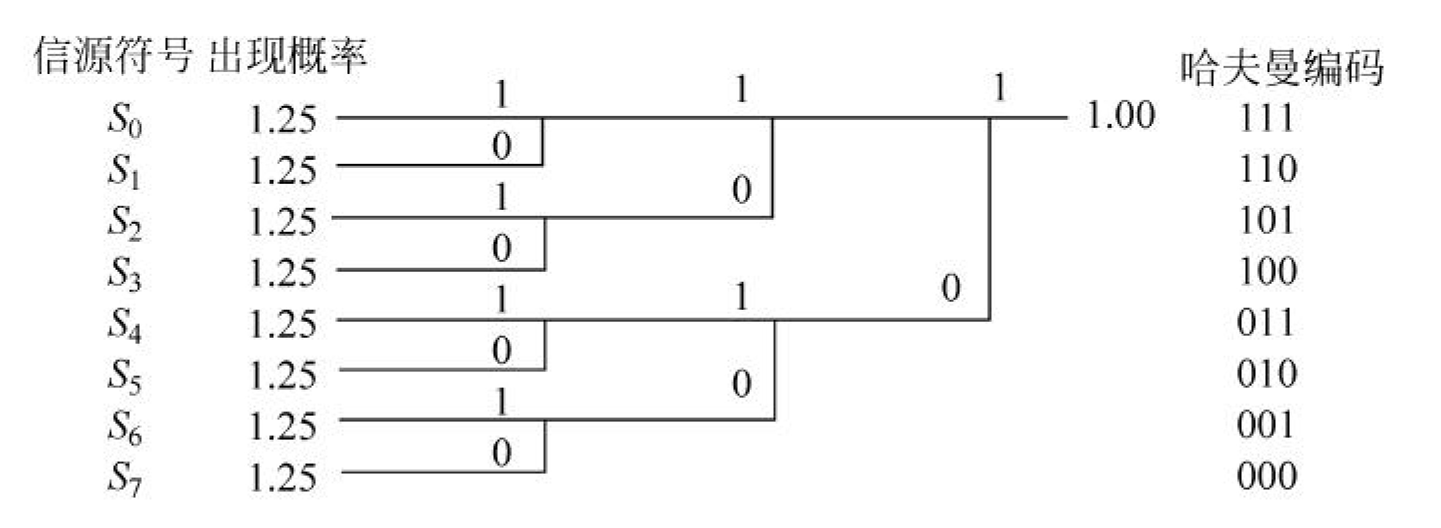


图 9-3 等概率分布霍夫曼编码过程

9.3.3 香农-范诺编码

香农—范诺(Shannon-Fannon)编码也是一种典型的可变字长编码。与霍夫曼编码相似,当信源符号出现的概率正好为 2 的负幂次方时,香农—范诺编码的编码效率可以达到 100%。

1. 编码步骤

香农—范诺编码的理论基础是符号的码字长度 N_i 完全由该符号出现的概率来决定,对于二进制编码即有

$$-\log_2 p_i + 1 \geqslant N_i \geqslant -\log_2 p_i \tag{9-11}$$

香农—范诺编码的实现步骤如下:

(1) 将信源符号按其出现的概率由大到小顺序排列,若两个符号的概率相等,则相等概率的字符顺序可以任意排列。

(2) 根据式(9-11)计算出各概率符号所对应的码字长度 N_i。

(3) 将各符号的概率累加,计算累加概率 P,即

$$\begin{cases} P_0 = 0 \\ P_1 = p_0 \\ P_2 = p_0 + p_1 \\ P_3 = p_0 + p_1 + p_2 \\ \cdots \\ P_i = p_0 + p_1 + p_2 + \cdots + p_{i-1} = P_{i-1} + p_{i-1} \end{cases} \tag{9-12}$$

(4) 把各个累加概率 P 由十进制转换为二进制。

(5) 根据式(9-11)取二进制累加概率前 N_i 位的数字,并省去小数点前的"0."字符,即为对应信源符号的香农—范诺编码码字。

2. 编码举例

例 9-2:香农—范诺编码示例

为便于与霍夫曼编码方法进行比较,本例仍以例 9-1 的图像数据进行香农—范诺编码。编码步骤如下:

(1) 将信源符号按其出现概率由大到小顺序排列,为 0.40、0.18、0.10、0.10、0.07、0.06、0.05、0.04。

(2) 对于概率 0.40 对应的符号 S_0,根据式(9-11)计算 $N_0=2$,将累加概率 0.00 转换位

二进制小数为 0.00，取前 $N_0=2$ 位，并去除小数点前的字符，即 S_0 字符编码为 00。

(3) 对于概率 0.18 对应的符号 S_1，根据式(9-11)计算 $N_1=3$，将累加概率 0.40 转换位二进制小数为 0.0110，取前 $N_1=3$ 位，并去除小数点前的字符，即 S_1 字符编码为 011。

(4) 对于概率 0.10 对应的符号 S_2，根据式(9-11)计算 $N_2=4$，将累加概率 0.58 转换位二进制小数为 0.10010，取前 $N_2=4$ 位，并去除小数点前的字符，即 S_2 字符编码为 1001。

(5) 对于概率 0.10 对应的符号 S_3，根据式(9-11)计算 $N_3=4$，将累加概率 0.68 转换位二进制小数为 0.10100，取前 $N_3=4$ 位，并去除小数点前的字符，即 S_3 字符编码为 1010。

(6) 对于概率 0.07 对应的符号 S_4，根据式(9-11)计算 $N_4=4$，将累加概率 0.78 转换位二进制小数为 0.11000，取前 $N_4=4$ 位，并去除小数点前的字符，即 S_4 字符编码为 1100。

(7) 对于概率 0.06 对应的符号 S_5，根据式(9-11)计算 $N_5=5$，将累加概率 0.85 转换位二进制小数为 0.1101100，取前 $N_5=5$ 位，并去除小数点前的字符，即 S_5 字符编码为 11011。

(8) 对于概率 0.05 对应的符号 S_6，根据式(9-11)计算 $N_6=5$，将累加概率 0.91 转换位二进制小数为 0.1110100，取前 $N_6=5$ 位，并去除小数点前的字符，即 S_6 字符编码为 11101。

(9) 对于概率 0.04 对应的符号 S_7，根据式(9-11)计算 $N_7=5$，将累加概率 0.96 转换位二进制小数为 0.11110100，取前 $N_7=5$ 位，并去除小数点前的字符，即 S_7 字符编码为 11110。

3. 香农—范诺编码效能

(1) 图像信息熵为

$$H=-\sum_{k=0}^{7} p_k \log_2 p_k = 2.55$$

(2) 平均码字长度为

$$\begin{aligned} R &= \sum_{k=1}^{K} B_k p_k \\ &= 2\times 0.40 + 3\times 0.18 + 4\times 0.10 + 4\times 0.10 + 4\times 0.07 + 5\times 0.06 + \\ &\quad 5\times 0.05 + 5\times 0.04 = 3.17 \end{aligned}$$

(3) 效率为

$$\eta=\frac{H}{R}\times 100\% = \frac{2.55}{3.17} = 80.4\%$$

(4) 信息冗余度为

$$\upsilon = 1-\eta = 19.6\%$$

结果如表 9-6 所示。

表 9-6　香农—范诺编码效能表

信源符号	出现概率	码字长度	累加概率	转换位二进制	香农—范诺码字
S_0	0.40	2	0	0.00	00
S_1	0.18	3	0.40	0.011	011
S_2	0.10	4	0.58	0.1001	1001
S_3	0.10	4	0.68	0.10100	1010
S_4	0.07	4	0.78	0.1100	1100
S_5	0.06	5	0.85	0.11011	11011
S_6	0.05	5	0.91	0.11101	11101

续表

信源符号	出现概率	码字长度	累加概率	转换位二进制	香农—范诺码字
S_7	0.04	5	0.96	0.111101	11110
编码效能	信息熵			$H=2.55$	
	平均码长			$R=3.17$	
	编码效率			$\eta=80.4\%$	

4. 香农—范诺编码的特点

香农—范诺编码具有以下特点：

(1) 与霍夫曼编码一样，香农—范诺编码属于熵编码，需要知道各信源字符出现的概率。

(2) 香农—范诺编码效率比霍夫曼编码效率略低一些。在这个例子中最后所得到的平均码长大于3bit，也就是说大于采用等长编码方法所需要的码长。这是因为这里所给的码字集合的个数太少而引起的，在一般情况下，香农编码的平均码长还是比均匀编码的码长短。

5. 香农—范诺编码的对分实现法

香农—范诺编码的另一种编码方法则不需要根据式(9-11)进行计算，直接根据各符号出现的概率序列进行编码，称为香农—范诺编码的对分法(二分法)实现法。具体实现步骤如下：

(1) 根据各符号出现的概率，按由大到小的顺序排列，如表9-7所示。

(2) 在该概率序列的某个中间位置将其对分，即分为两个子序列。对分的原则为尽量使两个子序列的概率之和近似相等，给上面的子序列编码为1，下面的子序列编码为0(编码见括号内的数字，也可以反过来)。

(3) 重复步骤(2)，直到各个子序列都被对分到只有一个元素为止。

(4) 将每个元素所属的子序列值从左到右依次编码，即为各符号对应的香农—范诺编码。

(5) 根据有关公式计算编码效能，结果如表9-7下面3行所示。

表9-7 香农—范诺编码效能表

信源符号	出现概率	对分				编码
S_0	0.40	0.58 (1)	0.40(1)	—	—	11
S_1	0.18		0.18(0)			10
S_2	0.10	0.42 (0)	0.20(1)	0.10(1)		011
S_3	0.10			0.10(0)		010
S_4	0.07		0.21(0)	0.13(1)	0.07(1)	0011
S_5	0.06				0.06(0)	0010
S_6	0.05			0.09(0)	0.05(1)	0001
S_7	0.04				0.04(0)	0000
编码效果	信息熵	$H=2.55$				
	平均码长	$R=2.64$				
	编码效率	$\eta=96.59\%$				

6. 对分法的特点

对分法具有以下特点：

(1) 根据表9-7所示结果，香农—范诺编码的对分实现方法的编码效率较之第一种实现方法略高，但比霍夫曼编码略低。

(2) 香农—范诺编码的字符码字长度并不完全遵守式(9-11)，事实上，香农—范诺对分法编码也没有采用式(9-11)进行计算。

9.3.4 行程长度编码

行程长度编码又称RLE压缩方法，RLE是英文Run Length Encoding的缩写，RLE编码属于熵编码，是最简单的图像压缩的方法之一，曾广泛地应用于各种图像格式文件的数据压缩处理。

1. RLE编码的原理

RLE编码的原理比较简单，即在被压缩文件中寻找连续重复的数值，以重复次数和重复数值自身两个值取代文件中的连续值，重复次数称为行程长度。例如，若有一串字符信息*aaaaabbbbcccddeefffaa*，经RLE编码处理可表示为*5a4b3c2d2e3f2a*。

RLE编码的特点简单明了，一般用于包含大量重复信息的数据时可以获得很好的压缩效率。但对于连续重复次数较少的数据文件，则难以获得较好的压缩比，甚至可能产生压缩后的编码字节数大于处理前的图像字节数。所以行程长度编码的压缩效率与图像数据的重复情况密切相关。

另外，由于利用应用程序实现行程长度编码的压缩时，分别以一个字节表示重复数据自身，用另一个字节表示数据重复次数，以这两个字节代替一串重复出现的数据。为区分不同数据的含义，通常需要在表示重复次数的字节的前一位或二位作为标志位，提示该字节不是数据本身而是连续重复的次数。因此，两个字节的编码最多只能代替63个(2^6-1)或127个(2^7-1)字节长度的重复数据。如果超出了所能表现的长度，则可以采取增加字节数的方式予以解决。

不同格式的图像文件，如BMP、PCX、TIFF等在采用行程长度编码压缩图像数据时的编码方法也略有不同。RLE编码比较适合于二值图像文件。

2. PCX文件的RLE编码方法

PCX格式的图像文件是采用行程压缩方法处理数据的。压缩编码原理是：在256色的PCX文件中，每个像素占一个字节，压缩数据以字节为单位逐行进行编码，每行填充到偶数字节数，若在图像数据中发现一串连续相同的数据，就以两个字节代替这串连续相同的数据存入图像文件中。这两个字节中的第一个字节表示这串相同数据的重复次数，第二字节代表重复数据的值。对于非重复数据则单独处理。另外第一字节的前两位作为标志位置为1，即0XC0 (11000000)，其余6位所表示的才是相同数据的个数，因此最大值为63。PCX格式文件编码时最大行程长度为63，如果连续相同的数据超过63个，则必须分两次表示。

例9-3：RLE编码方法示例

设有如下一串原始图像数据，以PCX格式文件进行压缩。

0X18　0X18　0X18 … 0X18(共12个) 0X5A　0X67　0X5F　0X71　0X69　0X28 … 0X28(共80个)0XD7　0XD9　0XCC　0XC8

根据 PCX 格式的图像文件的 RLE 编码原理，压缩编码如下：

```
0XCC  0X18  0X5A  0X67  0X5F  0X71  0X69  0XFF  0X28  0XD1  0X28
0XC1  0XD7  0XC1  0XD9  0XC1  0XCC  0XC1  0XC8
```

需要注意的是，对于大于或等于 0XC0(192)的数据，即使不重复出现，也必须以两个字节表示。因此，原始数据的后四个非重复数据，为避免解压缩数据时错误地将 0XD7 和 0XD9 解压为连续 23 个 0XD9，对每一个大于 0XC0 的数据采用两个字节表示。

还应指出的是，尽管现在几乎所有的图像应用软件都支持 PCX 文件格式，但由于压缩比不高，目前的应用不是很多。

3. RLE4 编码方法

用于 BMP 文件的压缩编码方法与 PCX 文件编码方法略有不同，以 16 色位图数据的 RLE4 压缩算法为例，RLE4 与一般的行程编码算法相同点是都以两个字节表示图像数据中一串重复的数据。其中第 1 字节表示数据重复次数，第 2 字节表示重复数据自身。不同点是 RLE4 压缩算法在第 1 字节中所表示的重复数据的次数是像素重复次数，而不是一般 RLE 编码中所定义的字节重复次数。

例如，一般 RLE 编码中的数据 0X08 0X58 表示将图像数据 0X58 连续重复显示 8 个字节，但在 RLE4 压缩算法中，则表示连续显示 8 个像素。另外，RLE4 压缩算法与一般 RLE 压缩算法不同点还表现在：由于是针对 16 色图像数据，如表 9-8 所示，引进了 4 组特殊的识别码。

表 9-8　RLE4 标识字符表

标识字符	含　义	用　法
0X00 0X00	表示该行图像数据已经结束	RLE4 编码的一个压缩单元为图像数据的一行，每个压缩单元结束时，都要引入该识别字符进行标记
0X00 0X01	表示整个图像数据结束	应用 RLE4 算法处理完图像中的所有数据时，都要引入该识别字符，作为图像数据的结束标志
0X00 0X02	转义字符，即随后的两个字节分别表示下一个像素从当前位置开始的水平与垂直位移	常用于在已经确定的背景画面上的某个区域内增加图像时，则必须在这组数据之前，加入该识别字符进行标记
0X00N(0) (03≤N≤0XFF)	表示从当前位置起，图像数据存在连续 N 个不同的值	在该标识码之后有 N/2 个字节用来存放这 N 个不同的数据。在 RLE4 压缩算法中，每个字节包含两个像素的颜色值，当连续显示时，第 1 个像素按字节高 4 位表示的颜色进行显示，第 2 个像素则按字节的低 4 位显示，直至依次显示完该标识码所标识的 N 个像素点

由表 9-8 可知，每组标识字符都由两个字节组成，第 1 字节值均为 0。BMP 文件规定压缩数据的字节数必须为偶数，而每行压缩数据的字节数则必须是 4 的倍数，如果不满足该条件，就必须用 0 补齐。因此，如果不同值的图像数据个数为奇数，则必须在这些图像数据的

尾部加入字节 0X00 使之成为偶数个数据。在读取 BMP 压缩文件过程中，也要考虑这一特点，判断是否包含多余的字节。

例 9-4：设有一幅原始图像数据如下，以 16 色 BMP 文件进行压缩。

第 1 行：0X28 0X28 0X28 0X28 0X30 0X60 0X58 0X09 0X22 … 0X46 … 0X46(共 8 个)

第 2 行：0X18 0X68 0X67 0X88 0X88 0X88 0X88 … 0X90 0X78

根据 16 色图像文件的 RLE4 编码原理，压缩后的图像数据为：

第 1 行：0X08 0X28 0X00 0X0a 0X30 0X60 0X58 0X09 0X22 … 0X00 0X10 0X46 0X00 0X00

第 2 行：0X00 0X06 0X18 0X68 0X67 0X00 0X08 0X88 … 0X00 0X04 0X90 0X78 0X00 0X01

9.3.5 算术编码

从理论上分析，采用霍夫曼编码可以获得最佳信源字符编码效果，但是实际应用中，由于信源字符出现的概率并非满足 2 的负幂次方，因此往往无法达到理论上的编码效率和信息压缩比。以信源字符序列$\{x,y\}$为例说明如下：

设字符序列$\{x,y\}$对应的概率为$\{1/3,2/3\}$，N_x 和 N_y 分别表示字符 x 和 y 的最佳码长，则根据信息论有

$$N_x = -\log_2\left(\frac{1}{3}\right) = 1.58 \tag{9-13}$$

$$N_y = -\log_2\left(\frac{2}{3}\right) = 0.588 \tag{9-14}$$

因此，字符 x、y 的最佳码长分别为 1.58bit 和 0.588bit。这表明，要获得最佳编码效果，需要采用小数码字长度。然而对于数字计算机而言，这是不可能实现的。只能按整数位进行，即采用霍夫曼方法对$\{x,y\}$的码字分别为 0 和 1，也就是两个符号信息的编码长度都为 1。对于出现概率大的字符 y 并未能赋予较短的码字。因此，实际编码效果往往不能达到理论效率。为提高编码效率，Elias 等提出了算术编码算法。

1. 算术编码的特点

算术编码是信息保持型编码，它不像霍夫曼编码，无须为一个符号设定一个码字。算术编码可以分为固定方式编码和自适应方式编码两种。选择不同的编码方式，将直接影响到编码效率。自适应算术编码的方式，无须先定义概率模型，适合于无法知道信源字符概率分布的情况。这也是算术编码优于霍夫曼编码的地方之一。同时，当信源字符出现的概率比较接近时，算术编码效率高于霍夫曼编码的效率，在图像通信中常用它来取代霍夫曼编码。不足之处是实现算术编码算法的硬件比霍夫曼编码复杂。

2. 编码原理

算术编码的方法是将被编码的信源消息表示成 0～1 之间的一个间隔，即小数区间，消息越长，编码表示它的间隔就越小，由于以小数表示间隔，因而表示的间隔越小所需的二进制位数就越多，码字就越长。反之，间隔越大，编码所需的二进制位数就少，码字就短。信源中连续符号根据某一模式生成概率的大小来缩小间隔，可能出现的符号要比不太可能出现

的符号缩小范围少，只增加了较少的比特。

算术编码将被编码的图像数据看作是由多个符号组成的字符序列，对该序列递归地进行算术运算后，成为一个二进制分数。在接收端，解码过程也是算术运算，由二进制分数重建图像符号序列。下面以一个实例来说明算术编码的基本原理。

3. 编码实例

例 9-5：算术编码示例

设图像信源编码可用 a、b、c、d 这 4 个符号来表示，若图像信源字符集为$\{dacba\}$，信源字符出现的概率分别如表 9-9 所示，采用算术编码对图像字符集编码。

表 9-9　信源字符出现概率

信源字符	a	b	c	d
出现概率	0.4	0.2	0.2	0.2

算术编码的基本步骤如下：

(1) 根据已知条件和数据可知，信源各字符在区间[0,1]内的子区间间隔分别如下：

$$a=[0.0,0.4) \qquad b=[0.4,0.6)$$
$$c=[0.6,0.8) \qquad d=[0.8,1.0)$$

(2) 计算中按如下公式产生新的子区间：

$$\begin{cases}\text{Start}_N=\text{Strat}_B+\text{Left}_C\times L\\ \text{End}_N=\text{Strat}_B+\text{Right}_C\times L\end{cases} \tag{9-15}$$

式中，Start_N、End_N 分别表示新子区间的起始位置和结束位置；Strat_B 为前一子区间的起始位置；Left_C、Right_C 分别表示当前符号字区间的左、右端；L 为前子区间的长度。

(3) 第 1 个被压缩的字符为 d，其初始子区间为[0.8,1.0)。

(4) 第 2 个被压缩的字符为 a，由于其前面的字符取值区间为[0.8,1.0)，因此，字符 a 应在前一字符区间间隔[0.8,1.0)的[0.0,0.4)子区间内，根据式(9-15)可得

$$\text{Start}_N=0.8+0.0\times(1.0-0.8)=0.8$$
$$\text{End}_N=0.8+0.4\times(1.0-0.8)=0.88$$

(5) 第 3 个被压缩的字符为 c，由于其前面的字符取值区间为[0.8,0.88)，因此，字符 c 应在前一字符区间间隔[0.8,0.88)的[0.6,0.8)子区间内，根据式(9-15)可得

$$\text{Start}_N=0.8+0.6\times(0.88-0.8)=0.848$$
$$\text{End}_N=0.8+0.8\times(0.88-0.8)=0.864$$

(6) 第 4 个被压缩的字符为 b，由于其前面的字符取值区间为[0.848,0.864)，因此，字符 b 应在前一字符区间间隔[0.848,0.864)的[0.4,0.6)子区间内，根据式(9-15)可得

$$\text{Start}_N=0.848+0.4\times(0.864-0.848)=0.8544$$
$$\text{End}_N=0.848+0.6\times(0.864-0.848)=0.8576$$

(7) 第 5 个被压缩的字符为 a，由于其前面的字符取值区间为[0.8544,0.8)，因此，字符 a 应在前一字符区间间隔[0.8544,0.8576)的[0.0,0.4)子区间内，根据式(9-15)可得

$$\text{Start}_N=0.8544+0.0\times(0.8576-0.8544)=0.8544$$
$$\text{End}_N=0.8544+0.4\times(0.8576-0.8544)=0.85568$$

(8) 经过上述计算,字符集{*dacba*}被描述在实数[0.8544,0.85568)子区间内,即该区间内的任一实数值都唯一对应该字符序列{*dacba*}。因此,可以用[0.8544,0.85568)内的一个实数表示字符集{*dacba*}。

(9) [0.8544,0.85568)子区间的二进制表示形式为[0.1101101010000110,0.1101101100001101),在该区间内的最短二进制代码为0.11011011,去掉小数点及其前的字符,从而得到该字符序列的算术编码为11011011。

如果所需要编码的字符信息更长,还可以依此类推地继续运算下去。随着被编码字符的增加,区间越来越小,则编码产生的二进制码越来越长。

算术编码可以通过硬件电路实现,在上述乘法运算中,可以通过右移来实现,因此在算术编码算法中只有加法和移位运算。

4. 算术编码效能

根据上述运算结果,编码11011011唯一代表字符序列{*dacba*},因此,平均码字长度为

$$R=\frac{8}{5}=1.6\text{bit/字符}$$

由此可得出,算术编码具有很高的编码效率,大量数据表明,算术编码一般比霍夫曼编码效能高5%以上。虽然该例给出的是一个二进制的算术编码的实例,它可以看作多进制算术编码的特例,多进制编码的原理与此类似。在H.263视频编码标准中,就是将算术编码作为一个选项来代替霍夫曼编码,以期提高VLC的效率。

9.4 位平面编码

位平面编码又称为比特平面编码,位平面编码是一种通过单独地处理图像的位平面来减少像素间冗余的有效技术。它将一幅多级图像分解为一系列二值图像并采用几种熟知的二值图像压缩方法对每一幅二值图像进行压缩。位平面编码分为两个步骤:位平面分解和位平面编码。

9.4.1 位平面二值分解

对于256灰度级(位深度为8)的灰度图像,每一个像素的灰度值可用8位二进制数表示,若灰度图像的每一个像素点均以二进制表示其灰度值,且均按从低位(b0)到高位(b7)的顺序排列二进制位,则取全部像素点的b0~b7,原图像可以形成8个二进制位平面图,如图9-4所示。

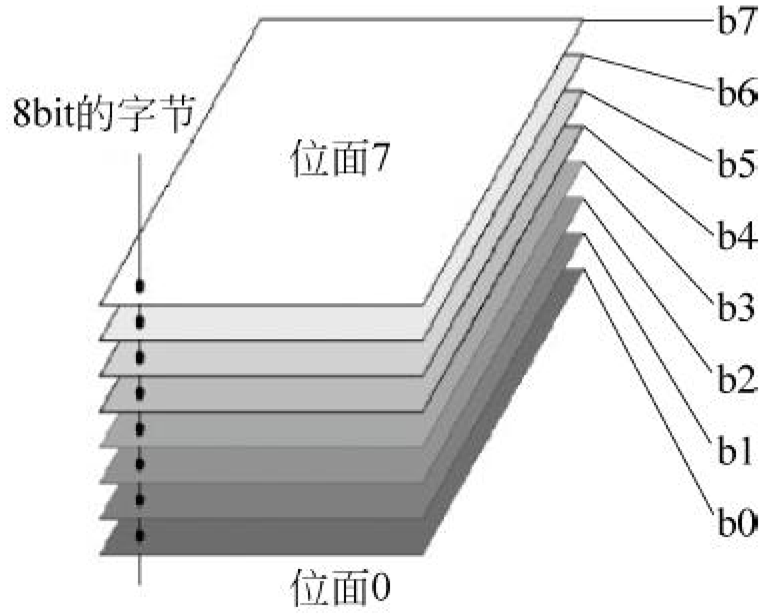


图9-4 灰度图像的位平面示意图

一幅n比特的灰度图像的灰度级可以用2为基的多项式表示为

$$a_{n-1}2^{n-1}+a_{n-2}2^{n-2}+\cdots+a_1 2^1+a_0 2^0 \quad (9\text{-}16)$$

式中,$a_i=0$或1。

根据上式,将多项式的n个系数分离到n个1bit的位平面中,这就实现了将一幅多级灰度图像表示成由n个二值图像组成的集合。0级位面就是原图像灰度级的第0位,也就是由上式从右往左中的第一位生

成的，$n-1$ 级位面就是原图像灰度级的第 $n-1$ 位，是由上式中的系数生成的，每个位面根据其系数按 0 到 $n-1$ 进行编号，位平面每个像素的取值等于原图像的每个像素点的灰度值基于上式所对应的多项式系数。

采用上述表达式分解理论上可导致一个应引起注意的问题，即对于某些特殊的灰度级，当灰度值发生增(减)变化时，将会使位平面产生显著的响应。例如，对于一个 8 位的灰度图像，当灰度从 127 过渡到 128 时，8 个位面的值从 01111111 变成 10000000，8 个位面的值全部发生了改变。

例 9-6：位平面分解示例

对如图 9-5(a)所示的图像进行位平面分解，图 9-5(b)～图 9-5(i)依次为最高位 7 至最低位 0 的位平面图。位平面二值码分解的 MATLAB 代码如下：

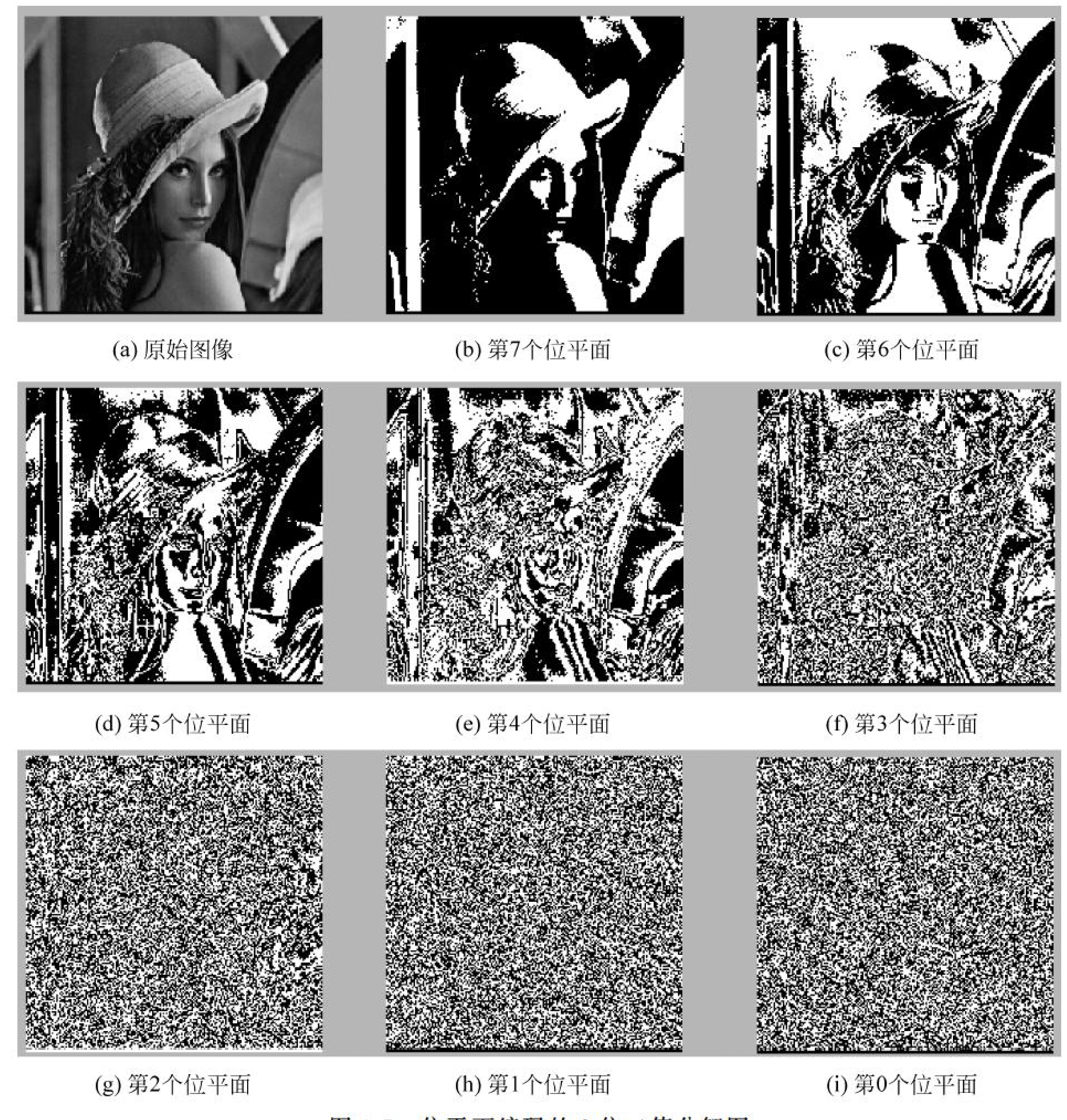
```
%灰度图像的二进制位平面分解代码
clc;
clear all;
X = double(imread('C:\Users\chen\Desktop\lena.bmp'));
[h w] = size(X);
subplot(3,3,1);
imshow(X,[]);
title('原始图像');
for k = 1:8
 for i = 1:h
    for j = 1:w
     %bitget 将 X(i,j)处灰度值分解为二进制串,然后取第 k 位
     tmp(i,j) = bitget(X(i,j),k);
    end
 end
 subplot(3,3,9 - k + 1);
 imshow(tmp,[]);
 ind = num2str(k - 1);
 imti = ['第',ind,'个位平面'];
 title(imti);
end
```

根据该例的实验结果，256 级灰度图像的位平面编码中，从图 9-5(b)～图 9-5(i)，即从高位到低位，视觉可见的有意义的信息逐渐减少，以前 5 位包含的图像信息最多，后三个位平面所包含的图像信息较少，主要是图像局部的小的细节信息，低位的位平面的甚至可以认为是噪声信息。

9.4.2　灰度码分解

为了避免位平面编码方法中二值分解法出现的这种某些灰度值稍微变化引起全部位平面变化的情况，可以采取先对灰度级进行编码。首先用一个 m 比特的灰度编码来表示图像，编码计算公式为

$$g_i=\begin{cases}a_i\oplus a_{i+1} & 0\leqslant i\leqslant n-1\\ a_i & i=n-1\end{cases}\tag{9-17}$$



(a) 原始图像　(b) 第7个位平面　(c) 第6个位平面

(d) 第5个位平面　(e) 第4个位平面　(f) 第3个位平面

(g) 第2个位平面　(h) 第1个位平面　(i) 第0个位平面

图 9-5　位平面编码的 8 位二值分解图

根据上式进行分解称为位平面的灰度码分解法，又称为格雷码，其结果仍是二值位平面图。由于上式采用了异或运算，使得连续码字之间仅有一个比特位置不相同，因此避免了某些特殊灰度值因小的变化引起整个位面产生变化。仍以前面提到的灰度值由 127 增 1，变为 128 为例，如果采用格雷码分解，则仅有第 7 个位平面从 0 变为 1，其他位平面没有发生变化，此时，对应 127 和 128 的格雷码分别为 01000000 和 11000000。

灰度图像经过分解之后，8 个位面的图像都是灰度值仅为 0 和 1 的二值图像，图像中存在着很多 0 或者 1 的连通区域，对这类二值图像通常可以采用行程长度编码(RLC)、常数块编码(CAC)等方法进行图像编码。

9.5　预测编码

预测编码是指依据某一模型，根据以往的样本值对于新样本值进行预测，然后将样本的实际值与预测值相减得到一个误差值，对这一误差值进行编码。如果模型足够好且样本序

列在时间和空间上存在较强的相关性，那么误差信号的幅度将远远小于信源原始信号，从而可以用较少的电平量对其差值量化得到较大的数据压缩结果。预测编码分为线性预测和非线性预测两类。

若能找到一个数学模型可以完全代表数据源，那么在接收端就能依据这一数学模型准确无误地产生出这些数据。但没有一个实际的系统能找到其完全准确的数学模型，因此，最好的办法是采用预测器以某种最小化的误差方法对下一个样本进行预测。

9.5.1 DPCM的工作原理

在线性预测中，最常用的是差分脉冲编码调制，即DPCM（differential pulse code modulation），Oliver在1952年对图像线性预测法进行了理论研究，1958年，Graham用计算机进行了DPCM模拟。DPCM的工作原理如图9-6所示，主要是基于图像中相邻像素之间的数据具有较强的相关性，每个像素可以根据以前已知的几个像素值进行预测。在DPCM编码中，编码和传输的不是像素值本身，而是这个取样值的预测值。

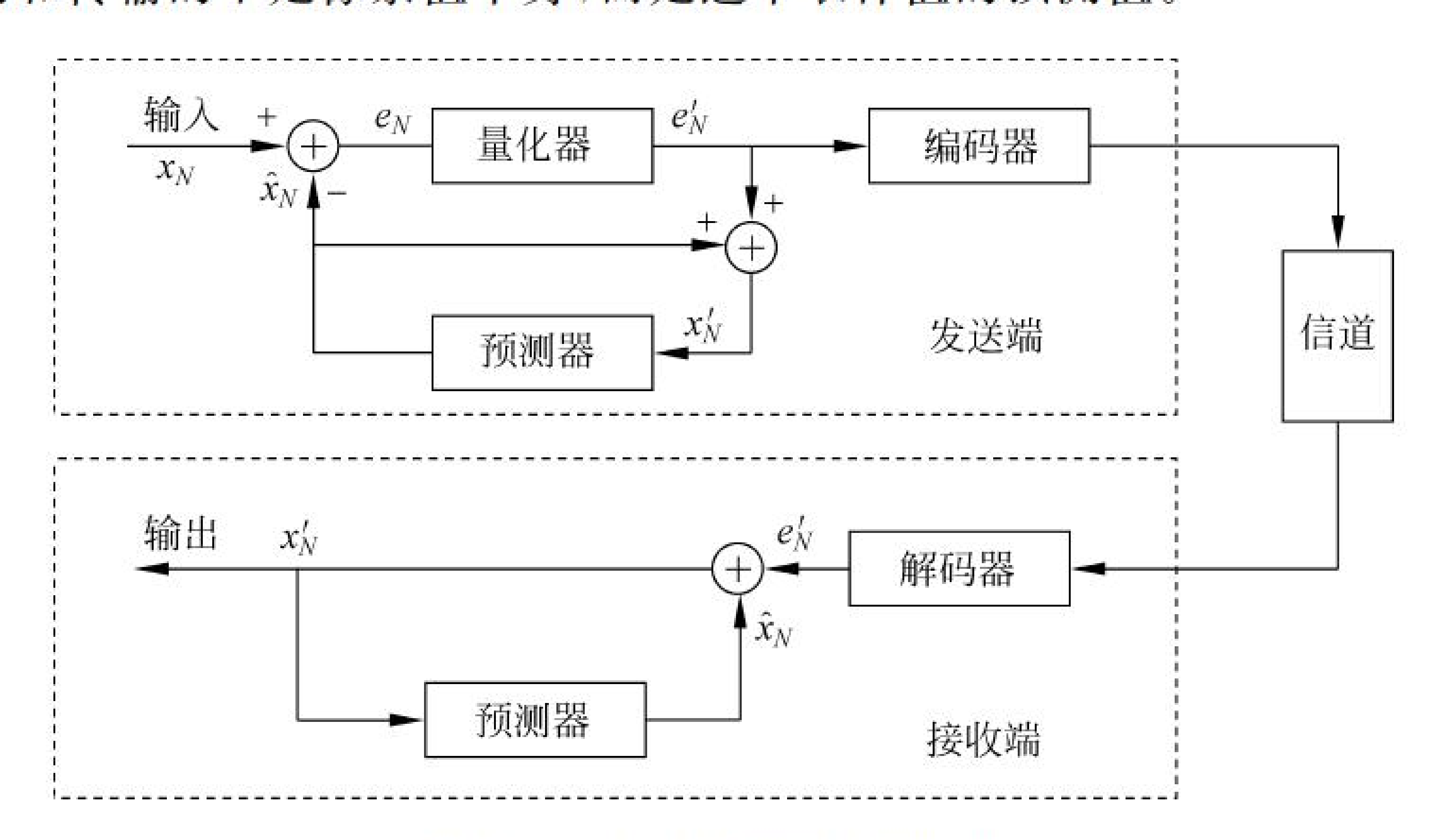


图9-6　DPCM系统原理框图

其中，x_N 为 t_N 时刻的亮度取样值；$\hat{x}_N$ 为预测器根据 t_N 时刻之前的样本值 $x_1, x_2, x_3, \cdots, x_{N-1}$ 对 x_N 所做的预测值；e_N 为差值信号，即

$$e_N = x_N - \hat{x}_N \tag{9-18}$$

量化器对 e_N 进行量化得到 $\hat{e}_N$，编码器对 $\hat{e}_N$ 进行编码。接收端解码时的预测过程与发送端相同，所采用的预测器也相同。接收端恢复的输出信号 x'_N 是 x_N 的近似值，两者的误差为

$$\Delta x_N = x_N - (\hat{x}_N - \hat{e}_N) = x_N - x'_N = e_N - \hat{e}_N \tag{9-19}$$

当 Δx_N 足够小时，输入信号 x_N 和DPCM系统的输出信号 x'_N 接近一致。

9.5.2 线性预测编码

在图像信源数据序列中，由 $x_1, x_2, x_3, \cdots, x_{N-1}$ 对 x_N 进行预测。由于是对 x_N 进行线性预测，因此，令 x_N 的预测值（估计值）为 $\hat{x}_N$，则 $\hat{x}_N$ 是 $x_1, x_2, x_3, \cdots, x_{N-1}$ 的线性组合。

设二维图像信号 $x(t)$ 是均值为零、方差为 σ^2 的平稳随机过程，$x(t)$ 在 $t_1, t_2, t_3, \cdots, t_{N-1}$ 时刻的抽样值分别为 $x_1, x_2, x_3, \cdots, x_{N-1}$，那么 t_N 时刻抽样的线性预测值为

$$\hat{x}_N = \sum_{i=1}^{N-1} a_i x_i = a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1} \tag{9-20}$$

式中，a_i 为预测系数，即待定常数。

若各 a_i 确定，则可以根据上式构成线性预测器。根据线性预测定义，$\hat{x}_N$ 应非常逼近 x_N，这就要求各 a_i 为最佳系数。采用均方误差最小的准则，可求得各最佳的系数。

现定义 x_N 的均方误差为

$$E\{[e_N]^2\} = E\{[x_N - \hat{x}_N]^2\} \tag{9-21}$$

为使 $E\{[e_N]^2\}$ 最小，对式(9-21)微分可得

$$\begin{aligned}\frac{\partial}{\partial a_i}E\{[e_N]^2\} &= \frac{\partial}{\partial a_i}E\{[x_N - \hat{x}_N]^2\} \\ &= \frac{\partial}{\partial a_i}E\{[x_N - (a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1})]^2\} \\ &= -2E\{[x_N - (a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1}) x_i]\}\end{aligned} \tag{9-22}$$

式中，$i=1,2,3,\cdots,N-1$

根据极值条件，可得如下 $N-1$ 个线性方程组：

$$\begin{cases} E\{[x_N - (a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1}) x_1\} = 0 \\ E\{[x_N - (a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1}) x_2\} = 0 \\ E\{[x_N - (a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1}) x_3\} = 0 \\ \cdots \\ E\{[x_N - (a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1}) x_{N-1}\} = 0 \end{cases} \tag{9-23}$$

该方程可表示为

$$E\{[x_N - (a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_{N-1} x_{N-1}) x_i]\} = 0$$
$$(i = 1,2,3,\cdots,N-1)$$

令 x_i 和 x_j 的协方差为

$$R_{ij} = E[x_i, x_j] \quad i,j = 1,2,3,\cdots,N-1 \tag{9-24}$$

则上式可以表示为

$$R_{Ni} = \sum_{k=0}^{N-1} a_{ki} R_{ki} = a_{1i} R_{1i} + a_{2i} R_{2i} + a_{3i} R_{3i} + \cdots + a_{N-1i} R_{N-1i} \tag{9-25}$$

若所有的协方差 R_{ij} 已知或可以测出时，则通过上式可计算出 $N-1$ 个预测系数 a_i。

综上所述，可以得出以下几点结论：

(1) 预测模型的复杂程度取决于线性预测中使用以前的样本数目，样本点越多，则预测器越复杂，最简单的预测仅使用前一个样本点，称为前值预测。

(2) 若采用 x 的同一行中 x_N 的若干已知像素样本值，如 $x_1, x_2, x_3, \cdots, x_{N-1}$ 来对 x_N 进行预测，则称为一维预测。

(3) 若采用同一行及前几行内的已知像素样本值来预测 x_N，则称为二维预测。

(4) 若采用的已知像素不仅是前几行的，而且还包括前几帧的，那么则称为三维预测。

图 9-7 所示是一个简单的、在 JPEG 无损编码方案中采用的实际预测器，它给出了静止图像的一个完整的二维预测器。它只考虑

B	*A*
C	*Y*

图 9-7 预测器的组成

临近3点 A、B 和 C 的值，例如，以 A、B 或 C 作为 Y 的预测值，则共有如表9-10所示的多种线性预测方法。

表9-10　Y 的预测值方法

预测方法	0	1	2	3	4	5	6	7
预测值 $\hat{Y}$	非预测	A	B	C	$A+B-C$	$A+\frac{B-C}{2}$	$B+\frac{A-C}{2}$	$\frac{A+B}{2}$

9.5.3　非线性预测

线性预测编码的基础和前提是将整个图像域视为一个平稳随机过程，其自相关系数与像素在图像信源数据域中的位置无关。而实际上，图像的起伏变化是始终存在的，被描述的像素和周围像素之间存在多种多样的关系。线性预测系数是一种近似条件下的常数，它忽略了不同像素之间的个性条件，因此存在一些不足。

非线性预测针对线性预测的不足，充分考虑了图像的统计特性和图像信源数据的个别变化情况，即力求使预测系数与图像的实际局部特性相一致，通过使预测系数随预测条件而变化，从而进一步提高压缩编码的性能。

9.6　图像的变换编码

变换编码主要包括DFT变换、K-L变换、WHT变换、DCT变换和小波变换编码等，变换编码因为其独特的编码效果，已经成为一种得到广泛应用的图像压缩编码方法。

9.6.1　变换编码的基本原理

图像变换编码的基本思想是将空域中描述的图像数据经过某种变换，如DFT变换、DCT变换、K-L变换等二维正交变换，转换到新的变换域中进行描述，在变换域中达到改变能量分布的目的，将图像能量在空间域的分散分布变为在变换域中的相对集中分布，从而实现对信源图像数据的有效压缩。

变换编码的基本流程如图9-8所示，图像数据经过某种变换、量化和编码(通常为变长编码)后由信道传输到接收端，接收端进行相反的处理，即解码、反量化以及逆变换，然后输出原图像数据。

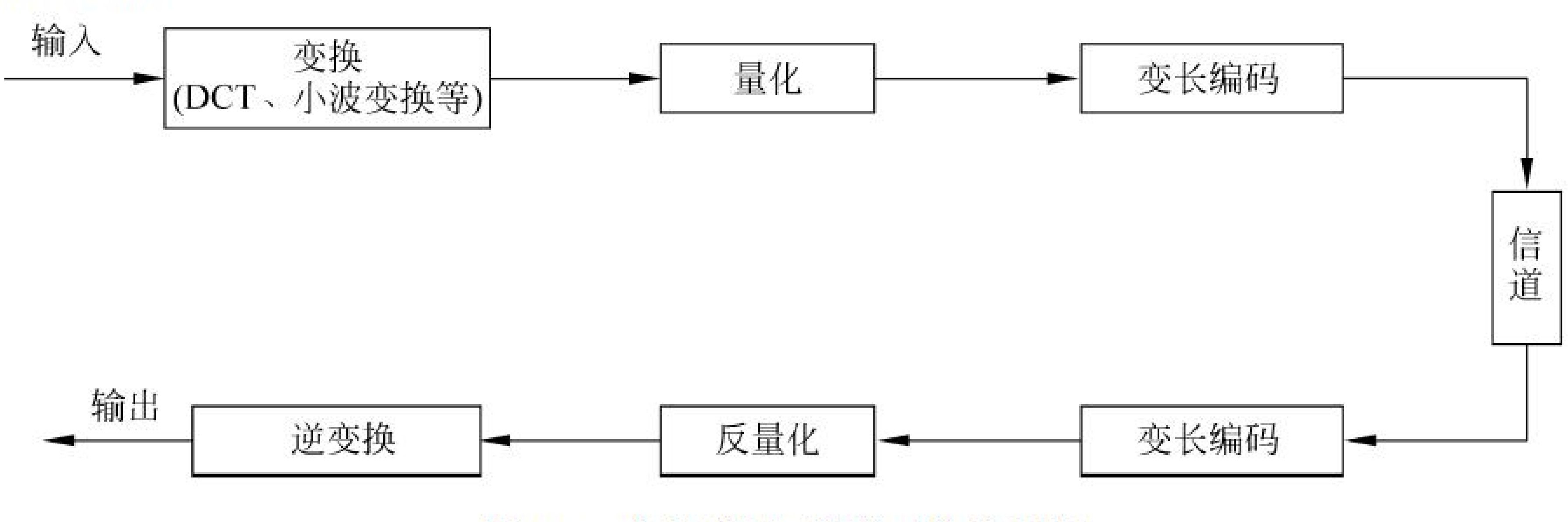


图9-8　变换编码、解码工作流程图

图像数据经过正交变换后，空域中的总能量在变换域中得到保持，但能量将会重新分布，并集中在变换域中少数的变换系数上，以达到压缩数据的目的。

9.6.2 正交变换的物理意义

通常情况下，变换编码都会选择正交变换，正交变换是一种数据处理手段，它将被处理的图像信源数据按照正交变换规则映射到另一个域进行处理。由于图像是以二维矩阵表示的，所以在图像编码中多采用二维正交变换形式。图像数据正交变换后不改变信源的熵值，变换前后图像的信息量没有损失，完全可以通过对应的逆变换得到原来的图像数据。但统计分析表明，经过正交变换后，数据的分布规律发生了很大的改变，像素之间的相关性下降，变换系数向新坐标系中的少数坐标集中，一般集中于少数的直流或低频分量的坐标点。变换编码将统计上高度相关的像素所构成的矩阵通过正交变换，变成统计上彼此较为独立、甚至达到完全独立的变换系数矩阵，以达到压缩数据的目的，这就是图像变换。

需要指出的是，如果将整个图像作为一个二维矩阵，则变换处理运算量太大，难以实现。所以在实用中，先将一幅图像分割成若干小的图像子块，如 8×8 或 16×16 小方块，各图像子块的像素值都可以看成为一个二维数据矩阵，变换是以这些图像子块为单位进行的。

如图 9-9 所示，x_1 和 x_2 分别表示两个像素的亮度取值，对于图 9-9(a)所表示的阴影区域而言，相邻像素之间存在相关性，绝大多数相邻两像素灰度值接近或相等，图中阴影部分表示像素 x_1 与 x_2 同时出现相近亮度值的可能性。x_1、x_2 的相关性越强，则阴影部分就更扁长，这时图像在 x_1、x_2 方向上的能量都较大。x_1、x_2 的相关性越弱，则阴影部分呈圆形状，说明 x_1 处于某一亮度值时，x_2 可能出现在不相同的任意亮度值上。

图 9-9(b)表示，若对该数据进行正交变换，几何上即相当于坐标系逆时针旋转 45°，成为 y_1 与 y_2 坐标系，这时阴影部分正好处于 y_1 轴附近，若阴影部分越长，则在 y_1 轴上的投影就越大，而在 y_2 轴上的投影就越小，这表明图像的能量都集中分布在 y_1 方向上。

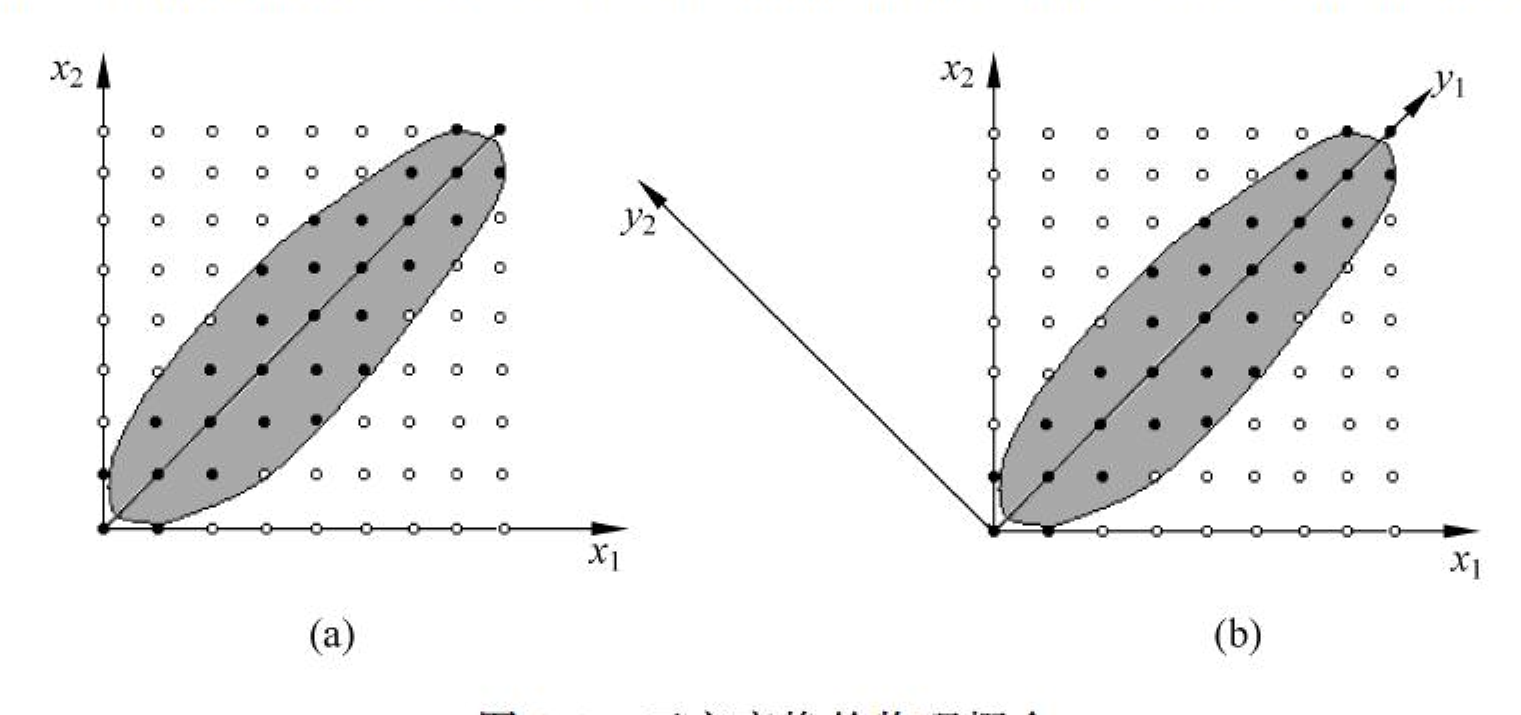


图 9-9 正交变换的物理概念

由图 9-9 可以看出，变换前 x_1 和 x_2 存在较大的相关性，能量分布集中在直线 $x_1 = x_2$ 附近。变换后，能量集中在 y_1 方向上，无论阴影部分在 y_1 轴上怎样变化，y_2 轴上的变化都非常小，这说明 y_1 与 y_2 的相关性减少，像素之间表现出独立性。因此，经过对图像信源数据进行正交变换，图像数据的相关性降低，通过在新的变换域中选择合理的量化方法就能达到压缩图像数据的目的。

9.6.3 变换类型与子块大小的选择

根据数字信号处理理论,所有正交变换中以 K-L 变换性能最优,经 K-L 变换后各变换系数在统计上不再相关,其协方差矩阵为对角阵,因而大大减少了图像原始数据的冗余度。因此,K-L 变换能完全消除图像子块内像素间的相关性,若舍弃一些特征值较小的变换系数,那么所引起的均方误差是所有正交变换中最小的。由于 K-L 变换是以原始图像各子块协方差矩阵的特征向量作为变换后的基向量,因此 K-L 变换的基对不同图像是不同的,与编码对象的统计特性有关,这种变换基的不确定性使得 K-L 变换在应用中不方便。因此,尽管 K-L 变换具有许多主要优点,但一般只用于进行理论上的比较。

DFT 变换是应用最早且非常成熟的变换之一,性能接近于最佳,且具有快速算法,其不足之处在于图像子块的变换系数在边界处不连续而造成恢复后的图像子块边界也不连续(即存在 Gibbs 现象),于是由子图像构成的整幅图像将呈现隐约可见的以图像子块为形状的小块状结构,影响了图像质量,一定程度上影响了其应用。

DCT 变换是图像变换中应用得最多的变换编码,其性能接近于 K-L 变换,且变换矩阵与图像内容无关。根据 DCT 变换的特点,还可避免 DFT 变换中图像子块边界处产生的跳跃与 Gibbs 现象。此外,市场上拥有许多基于 DCT 快速算法的 ASIC 芯片,因此,DCT 变换已经成为图像变换编码的主流。目前,JPEG、MPEG、H. 263 等国际编码标准都选择采用 DCT 变换模块。

沃尔什变换与 DCT 变换相比,其算法简单,因而运算速度较快,适用于高速实时系统,而且实现该算法的硬件结构简单,不足之处是性能比 DCT 变换差。

确定变换方式之后,还需要选择变换块的大小。由于压缩的依据是基于子块内图像像素间的相关性,若子块选得太小,不利于压缩比的提高。理论上,子块越大,计入的相关像素就越多,压缩比就越大。但如果子块过大,则计算量太大,同时考虑到距离较远的像素间相关性并不高,实际上过大的子块对压缩比的提高效果反而不好。因此,图像变换一般选择采用 8×8 或 16×16 大小的子块。

9.6.4 变换编码的步骤

根据图像变换编码的原理以及图 9-8 所示的编码、解码逻辑流程,实现变换编码一般包含以下步骤。

1. 原始图像分块

根据编码的具体要求,将图像划分为若干 $N\times N$ 的子块,即

$$\boldsymbol{X}=\begin{bmatrix} x_{00} & x_{01} & \cdots & x_{0(N-1)} \\ x_{10} & x_{11} & \cdots & x_{1(N-1)} \\ x_{20} & x_{21} & \cdots & x_{2(N-1)} \\ \cdots & \cdots & \cdots & \cdots \\ x_{(N-1)0} & x_{(N-1)1} & \cdots & x_{(N-1)(N-1)} \end{bmatrix} \tag{9-26}$$

通常情况下 N 取值为 8 或 16。图像分块之后,应同时根据编码的性能要求,综合考虑相关要素,选择变换矩阵 $\boldsymbol{A}$ 对各图像子块进行相应的正交变换。

设 $\boldsymbol{Y}$ 表示变换域中的图像数据,则可表示为

$$Y = AX \tag{9-27}$$

2. 变换域采样

即对变换域中的系数进行取舍。

3. 系数量化

由于变换之后的系数是不相关的，因此具有更大的独立性和有序性，利用量化使图像数据得到压缩。量化是产生有损压缩的原因，因此应选择合适的量化方法，以使量化失真最小。均方误差是衡量各种变换编码效能的一个重要准则，该准则可在较高的压缩比和一定的允许失真度之间寻求一个较理想的、可用的变换编码方式。

均方误差定义为

$$e = E\left[\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(y_{ij} - \hat{y}_{ij})^2\right] \tag{9-28}$$

式中，$\hat{y}_{ij}$ 为 y_{ij} 的量化值。

20 世纪 50 年代期间，Panter、Dire 和 Max 研究了使单个系数均方误差极小化的量化方案。研究发现，如果 y_{ij} 的概率密度函数是均匀的，那么具有均匀间隔输出的量化器是最佳的。对于其他的分布，使用非均匀量化器则能够起到减小均方差的作用。

4. 解码与反变换

在变换编码系统的接收端对所接收的比特流进行解码，分离出各变换系数 $\hat{y}_{ij}$，并进行系数的舍入，被舍弃的系数均以 0 代替，并进行逆变换运算，恢复各图像子块及整幅图像。

9.7 JPEG 标准

JPEG 是联合图片专家组的简称(joint photographic experts group)，JPEG 专门致力于对静止图像压缩编码技术的研究，该组织于 1991 年 3 月公布了对静止彩色图像和灰度图像(也称为连续色调图像)的压缩标准——JPEG 标准。JPEG 是面向静态图像编码的国际标准。在相同图像质量条件下，JPEG 文件拥有比其他图像文件格式更高的压缩比，JPEG 标准于 1992 年正式通过并作为 ISO/IEC 10918 标准公布。

9.7.1 JPEG 基本系统

JPEG 目前被广泛应用于多媒体和网络程序中，是现今万维网中使用最广泛的两种图像文件格式之一。JPEG 是一种有损压缩，即在压缩过程中会丢失数据，每次编辑 JPEG 图像后，图像就会被重新压缩一次，图像信息损失会有所增加。

JPEG 根据不同的应用场合对图像的压缩要求提出了几种不同的编码和解码方法，主要可以分为以下三个层次的系统：

(1) 基本系统。

(2) 扩展系统。

(3) 无损压缩系统。

所有符合 JPEG 标准的编解码器都必须支持基本系统，而其他系统则作为不同应用目的的可选项，即一个符合 JPEG 标准的编解码器至少应满足基本系统的技术指标。JPEG 标准还包括将图像质量控制在可视保真度高的范围内，可以设置压缩或质量水平，同时编码

器可被参数化；压缩标准可以应用于任何一类连续色调数字图像，并不应受到维数、颜色、画面尺寸、内容、影调的限制；压缩标准必须从完全无损到有损范围内可选，以适应不同的存储、CPU 和显示要求。JPEG 允许如下 4 种编码模式：

(1) 基本顺序模式编码。从左到右、从上到下对图像顺序进行基于 DCT 变换编码，虽然理论上 DCT 变换是可逆的，但由于计算时存在误差，实际上基于 DCT 的编码模式是一种有损编码。

(2) 渐进式 DCT 方式编码。基于 DCT 方式编码，对图像分层次进行处理，从模糊到清晰地传输图像(类似于 GIF 文件的交错方式)，具有频谱选择和逐次逼近两种实现方法。

① 频谱选择法：按 Z 形扫描的序号将 DCT 量化序数分成几个频段，每个频段对应一次扫描，每块均先传送低频扫描数据，得到原图概貌，再依次传送高频扫描数据，使图像逐渐清晰。

② 逐次逼近法：即每次扫描全部 DCT 量化序数，但每次的表示精度逐渐提高。

(3) 预测无损编码。使用线性预测器，如 DPCM 等，而不是基于 DCT。

(4) 分层模式编码。在空间域将信源图像以不同的分辨率表示，每个分辨率对应一次扫描，处理时可以基于 DCT 或预测编码，可以是渐进式，也可以是顺序式。

9.7.2　JPEG 编码方法

基本的 JPEG 算法属于变换类编码，基于 DCT 顺序式的基本系统编码流程如图 9-10 所示。

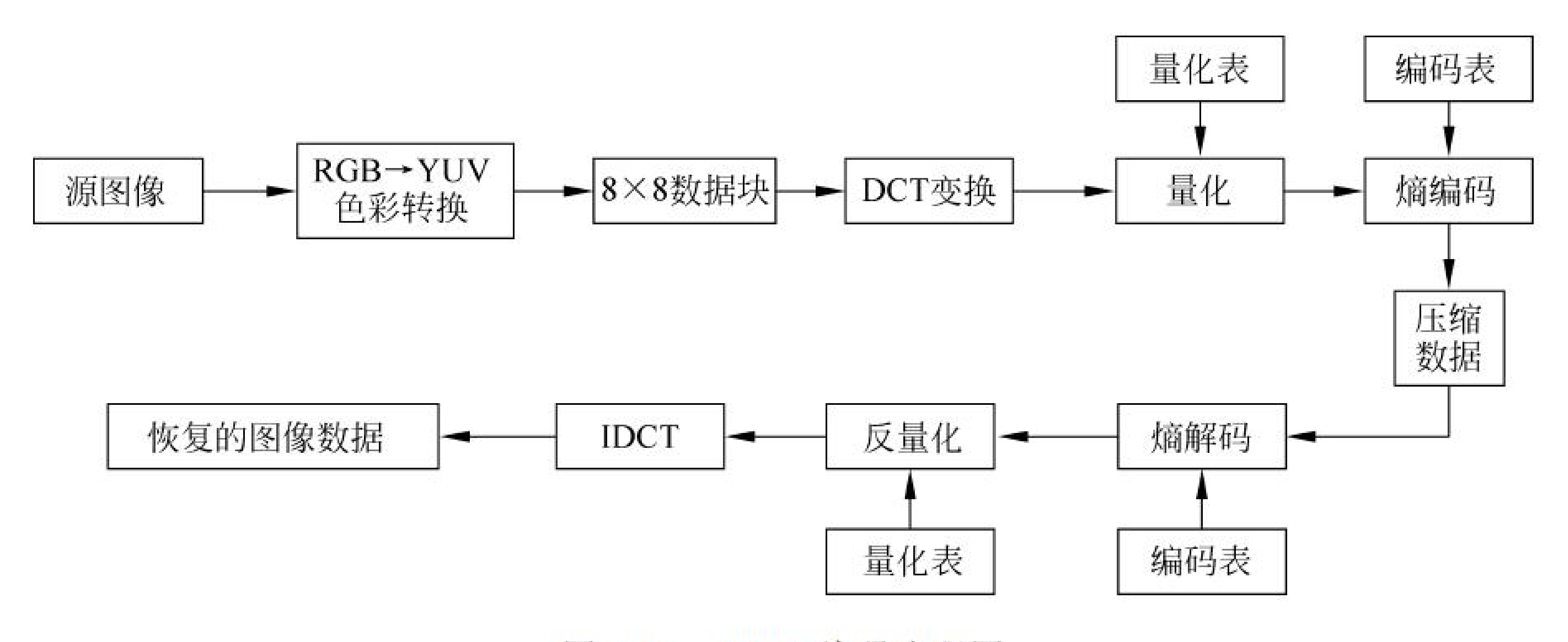


图 9-10　JPEG 编码流程图

1. 颜色空间转换

虽然 JPEG 标准可以压缩通常的 RGB 分量，但在 YC_bC_r 空间的压缩效果会更好，其中 Y 代表亮度，C_b、C_r 代表色度。这是因为人的视觉对色彩的变化不如对亮度的变化敏感，因而对色彩的编码可以比对亮度的编码粗糙些，这主要体现在不同的采样频率和量化精度上。在 JPEG 系统中，将彩色图像进行数据压缩之前必须对颜色模式进行转换，将 RGB 模式转为 YC_bC_r 模式。转换公式为

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ C_b = -0.169R - 0.3313G + 0.5B \\ C_r = 0.5R - 0.4187G - 0.0813B \end{cases} \tag{9-29}$$

将颜色空间转换后进行数据采样，由于亮度比色彩更重要，因而对 Y 分量的采样频率可高于对 C_b、C_r 的采样频率，这样有利于节省处理系统的存储空间，常用的采样方案有 YUV422 和 YUV411，即采样比例为 4∶2∶2 或 4∶1∶1 方式（对灰度图像而言，只有一个Y 分量）。这样的采样方式是基于视觉对亮度要比对色度更敏感，而重建后的图像与原图的差异是人的视觉所不易察觉到的。经过采样后的图像数据，色度数据比原来减少了一半。

2. 数据分块

对图像进行编码前，将每个分量图像分割成不重叠的图像子块，对于 JPEG 标准的 DCT 变换应选择分为 8×8 像素子块，每一个 8×8 像素子块称为一个数据单元。

3. DCT 变换

图像数据块分割为 8×8 子块之后，即以 8×8 子块为单位顺序进行二维 DCT 变换。对以无符号数表示的具有 p 位（一般用 8 位表示一个像素的颜色分量）精度的输入数据，在 DCT 前要减去 2^{p-1}，转换成有符号数，即在$[0,2^{p-1}]$范围内表示的无符号整数变成$[-2^{p-1},2^{p-1}-1)$范围内表示的有符号数，作为 IDCT 变换的输入量。在 IDCT 后应加上 2^{p-1}，转换成无符号数。对每个 8×8 的数据块进行 DCT 变换后，所获得 64 个变换系数代表了该图像子块的频率成分，其中低频分量集中在左上角，高频分量分布在右下角。系数矩阵左上角的系数称为直流（DC）系数，它代表了该数据块的平均值，其余 63 个系数为交流（AC）系数。

4. 系数量化

系数量化的目的是在保证图像质量的前提下，丢掉那些对视觉影响不大的信息。在 DCT 变换所得到的 64 个系数中，低频分量主要包含图像亮度等信息。在从空间域到频域的变换中，图像中的缓慢变化比快速变化更易引起视觉效果，所以在重建图像时，低频分量的重要性高于高频分量。因而在编码时可以忽略高频分量，从而达到压缩数据的目的。

JPEG 标准采用具有 64 个独立元素的量化表来实现 DCT 相应的 64 个系数的量化精度，使得对某个系数的具体量化与人类视觉对该频率分量的视觉敏感程度一致。理论上，对不同的空间分辨率、数据精度等情况，应该有不同的量化表。但一般采用图 9-11(a)对 Y 分量进行量化，而采用图 9-11(b)对色度分量进行量化，可取得较好的视觉效果。两张量化表充分考虑了 Y 分量比 C_b 和 C_r 更重要等因素，即对 Y 采用细量化，而对 C_b 和 C_r 采用粗量化。即以 DCT 变换后的系数除以量化表中相对应的量化参数，然后以四舍五入的方式取整，量化公式为

$$F_q(u,v) = \text{IntegerRound}\left[\frac{F(u,v)}{Q(u,v)}\right] \tag{9-30}$$

式中，$F_q(u,v)$表示量化以后的 DCT 系数；$F(u,v)$表示直接经 DCT 变换后的系数；$Q(u,v)$表示图 9-11 中的量化参数。

16	11	10	16	24	40	51	61
12	12	14	19	26	58	69	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
79	92	95	98	112	100	103	99

(a) 亮度量化表

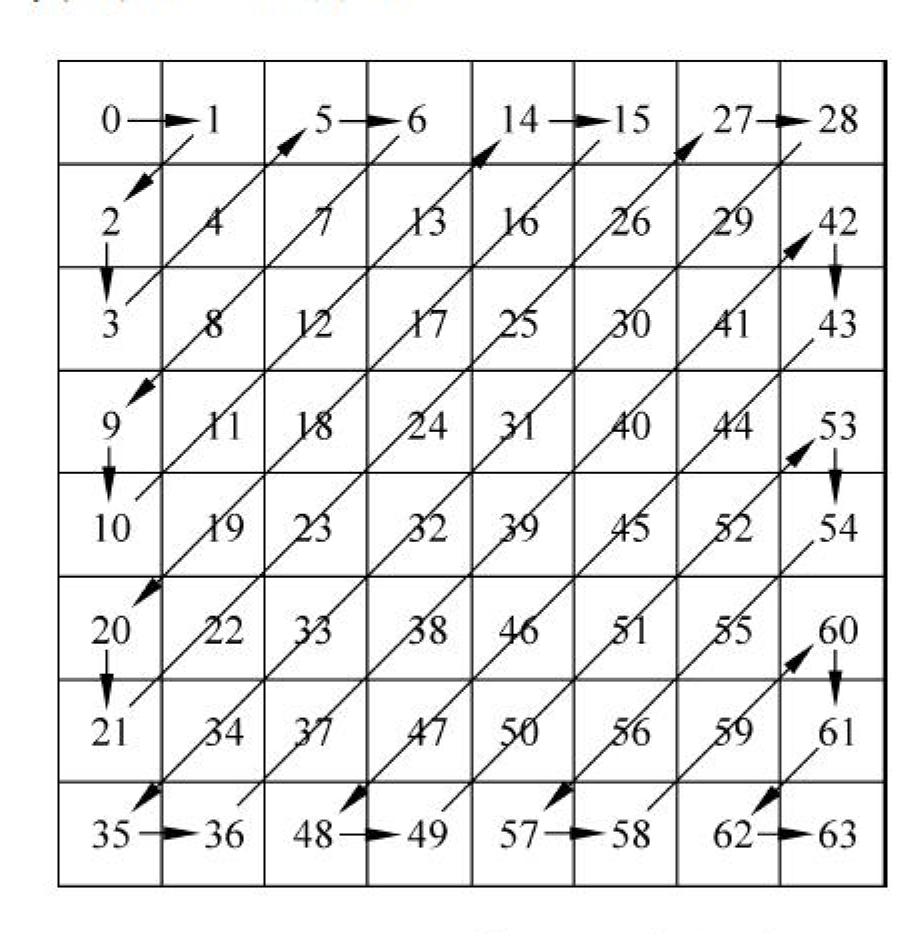
17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b) 色度量化表

图 9-11　亮度与色度量化表

5. Z 形扫描

DCT 系数量化后，构成一个稀疏矩阵，用 Z(Zigzag)形扫描将其变成一维数列，将有利于熵编码。Z 形扫描的顺序如图 9-12 所示。



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

图 9-12　AC 系数的 Z 形扫描表

6. DC 系数编码

DCT 变换产生的 64 个系数经过量化后，其中 $F(0,0)$为直流系数 DC，其余的 63 个为交流系数 AC，DC 系数的大小反映了一个 8×8 数据块的平均亮度。由于 8×8 的相邻子块之间 DC 系数有很强的相关性，所以 JPEG 对 DC 系数采用差分编码，即以前一数据块的同一分量的 DC 系数作为当前块的预测值，再对当前块的实际值与预测值的差值($DC_i - DC_{i-1}$)进行霍夫曼编码或算术编码。

若 DC 系数的动态范围为－1024～1024，则差值的动态范围可达－2047～＋2047，由于差分值范围太大，JPEG 没有采用对每一个差分值赋予一个码字，而是对码表进行了简化，采用"前缀码(SSSS)＋尾码"表示。前缀码指明了尾码的有效位数 B，可以根据差分值从表 9-11 中查出前缀码对应的霍夫曼编码。尾码的取值取决于 DC 系数的差值和前缀码。如果 DC 系数的差分值(D)大于等于 0，则尾码的码字为差分值的原码；否则，取差分值的 B 位反码。DC 系数编码的具体步骤如下：

(1) 对差分值进行判断，如果差分值大于 0，将差分值转换为二进制表示，并获得差值所占的位数。例如差值为 5 时，其二进制表示为 101，差分值的位数为 3。如果差分值小于 0，则取绝对值后转换为二进制码表示，再获得该编码值的反码。例如差值为－5 时，最后得反码为 010，差分值的位数为 3。

(2) 根据"前缀码(SSSS)＋尾码"的表示方法，以差分值所占的位数为索引值查表获得该差分值的编码。表 9-11 所示为亮度系数(DC)的霍夫曼编码表。查表可获得差分值为 5 时编码为 100，因此，结合前缀码(SSSS)和尾码，最终的编码值为 100101，而差值为－5 时的最终编码为 100010。表 9-11 表明，DC 系数差分值的最大位数为 11 位，即最大差值在 $(-2^{11}, 2^{11}-1)$范围内。

表 9-11　DC 系数差分值霍夫曼编码转换表

SSSS	DC 系数差值(D)	亮 度 码 字	色 度 码 字
0	0	00	00
1	−1,1	010	01
2	−3,−2,2,3	011	10
3	−7～−4,4～7	100	110
4	−15～−8,8～15	101	1110
5	−31～−16,16～31	110	11110
6	−63～−17,17～63	1110	111110
7	−127～−64,64～127	11110	1111110
8	−255～−128,128～255	111110	11111110
9	−511～−256,256～511	1111110	111111110
10	−1023～−512,512～1023	11111110	1111111110
11	−2047～−1023,1023～2047	111111110	11111111110

7. AC 系数编码

JPEG 基本系统对 63 个交流系数 AC 则采用行程长度编码,采用 Z 字形扫描是为了增加 0 的行程长度,从而更有利于压缩数据。交流系数 AC 的编码也可采用霍夫曼编码或算术编码。JPEG 在对 AC 系数进行 Z 行程长度编码后,将一个非零 AC 系数及其前面的 0 行程长度称为一个组合,记为[NNNN/SSSS],将每个组合以“[NNNN/SSSS]+尾码”进行编码表示,其中,NNNN 为 0 行程的长度,SSSS 表示尾码的有效位数,即当前非 0 系数所占的比特数,如果非零 AC 系数大于等于 0,则尾码的码字为该系数的原码,若 AC 系数小于 0,则以该系数绝对值的反码表示。

由于 JPEG 以 4 位表示 0 的行程长度,故在 JPEG 编码中,0 的最大行程只能等于 15。当 0 的行程长度大于 15 时,需要将其分开进行多次编码,即对前面的每 16 个 0 以“F/0”表示,对剩余 0 行程继续编码。

根据非零系数的数值可从表 9-12 查出对应的 SSSS,然后根据 NNNN/SSSS 从表 9-13 中查得其对应的霍夫曼编码。表 9-13 所示是 ITU-T 亮度 AC 系数的霍夫曼编码表,从表 9-13 可以看出,亮度 AC 系数差值的最大位数为 10 位,即最大差分值在$[-2^{10},2^{10}-1]$范围之内。

表 9-12　AC 系数的霍夫曼尾码位数表

SSSS	AC 系数的幅度
0	0
1	−1,1
2	−3,−2,2,3
3	−7～−4,4～7
4	−15～−8,8～15
5	−31～−16,16～31
6	−63～−17,17～63
7	−127～−64,64～127
8	−255～−128,128～255
9	−511～−256,256～511
10	−1023～−512,512～1023

表 9-13 亮度 AC 的霍夫曼编码表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
0/0(EOB)	4	1010	2	00
0/1	2	00	2	01
0/2	2	01	3	100
0/3	3	100	4	1010
0/4	4	1011	5	11000
0/5	5	11010	5	11001
0/6	7	1111000	6	111000
0/7	8	11111000	7	1111000
0/8	10	1111110110	9	111110100
0/9	16	1111111110000010	10	1111110110
0/A	16	1111111110000011	12	111111110100
1/1	4	1100	4	1011
1/2	5	11011	6	111001
1/3	7	1111001	8	11110110
1/4	9	111110110	9	111110101
1/5	11	11111110110	11	11111110110
1/6	16	1111111110000100	12	111111110101
1/7	16	1111111110000101	16	1111111110001000
1/8	16	1111111110000110	16	1111111110001001
1/9	16	1111111110000111	16	1111111110001010
1/A	16	1111111110001000	16	1111111110001011
2/1	5	11100	5	11010
2/2	8	11111001	8	11110111
2/3	10	1111110111	10	1111110111
2/4	12	111111110100	12	111111110110
2/5	16	1111111110001001	15	111111111000010
2/6	16	1111111110001010	16	1111111110001100
2/7	16	1111111110001011	16	1111111110001110
2/8	16	1111111110001100	16	1111111110001110
2/9	16	1111111110001101	16	1111111110001111
2/A	16	1111111110001110	16	1111111110010000
3/1	6	111010	5	11011
3/2	9	111110111	8	11111000
3/3	12	111111110101	10	1111111000
3/4	16	1111111110001111	12	111111110111
3/5	16	1111111110010000	16	1111111110010001
3/6	16	1111111110010001	16	1111111110010010
3/7	16	1111111110010010	16	1111111110010011
3/8	16	1111111110010011	16	1111111110010100
3/9	16	1111111110010100	16	1111111110010101
3/A	16	1111111110010101	16	1111111110010110

续表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
4/1	6	111011	6	111010
4/2	10	1111111000	9	111110110
4/3	16	1111111110010110	16	1111111110010111
4/4	16	1111111110010111	16	1111111110011000
4/5	16	1111111110011000	16	1111111110011001
4/6	16	1111111110011001	16	1111111110011010
4/7	16	1111111110011010	16	1111111110011011
4/8	16	1111111110011011	16	1111111110011100
4/9	16	1111111110011100	16	1111111110011101
4/A	16	1111111110011101	16	1111111110011110
5/1	7	1111010	6	111011
5/2	11	11111110111	10	1111111001
5/3	16	1111111110011110	16	1111111110011111
5/4	16	1111111110011111	16	1111111110100000
5/5	16	1111111110100000	16	1111111110100001
5/6	16	1111111110100001	16	1111111110100010
5/7	16	1111111110100010	16	1111111110100011
5/8	16	1111111110100011	16	1111111110100100
5/9	16	1111111110100100	16	1111111110100101
5/A	16	1111111110100101	16	1111111110100110
6/1	7	1111011	7	1111001
6/2	12	111111110110	11	11111110111
6/3	16	1111111110100110	16	1111111110100111
6/4	16	1111111110100111	16	1111111110101000
6/5	16	1111111110101000	16	1111111110101001
6/6	16	1111111110101001	16	1111111110101010
6/7	16	1111111110101010	16	1111111110101011
6/8	16	1111111110101011	16	1111111110101100
6/9	16	1111111110101100	16	1111111110101101
6/A	16	1111111110101101	16	1111111110101110
7/1	8	111110101	7	1111010
7/2	12	111111110111	11	11111111000
7/3	16	1111111110101110	16	1111111110101111
7/4	16	1111111110101111	16	1111111110110000
7/5	16	1111111110110000	16	1111111110110001
7/6	16	1111111110110001	16	1111111110110010
7/7	16	1111111110110010	16	1111111110110011
7/8	16	1111111110110011	16	1111111110110100
7/9	16	1111111110110100	16	1111111110110101
7/A	16	1111111110110101	16	1111111110110110
8/1	9	111111000	8	11111001

续表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
8/2	15	111111111000000	16	1111111110110111
8/3	16	1111111110110110	16	1111111110111000
8/4	16	1111111110110111	16	1111111110111001
8/5	16	1111111110111000	16	1111111110111010
8/6	16	1111111110111001	16	1111111110111011
8/7	16	1111111110111010	16	1111111110111100
8/8	16	1111111110111011	16	1111111110111101
8/9	16	1111111110111100	16	1111111110111110
8/A	16	1111111110111101	16	1111111110111111
9/1	9	111111001	9	111110111
9/2	16	1111111110111110	16	1111111111000000
9/3	16	1111111110111111	16	1111111111000001
9/4	16	1111111111000000	16	1111111111000010
9/5	16	1111111111000001	16	1111111111000011
9/6	16	1111111111000010	16	1111111111000100
9/7	16	1111111111000011	16	1111111111000101
9/8	16	1111111111000100	16	1111111111000110
9/9	16	1111111111000101	16	1111111111000111
9/A	16	1111111111000110	16	1111111111001000
A/1	9	111111010	9	111111000
A/2	16	1111111111000111	16	1111111111001001
A/3	16	1111111111001000	16	1111111111001010
A/4	16	1111111111001001	16	1111111111001011
A/5	16	1111111111001010	16	1111111111001100
A/6	16	1111111111001011	16	1111111111001101
A/7	16	1111111111001100	16	1111111111001110
A/8	16	1111111111001101	16	1111111111001111
A/9	16	1111111111001110	16	1111111111010000
A/A	16	1111111111001111	16	1111111111010001
B/1	10	1111111001	9	111111001
B/2	16	1111111111010000	16	1111111111010010
B/3	16	1111111111010001	16	1111111111010011
B/4	16	1111111111010010	16	1111111111010100
B/5	16	1111111111010011	16	1111111111010101
B/6	16	1111111111010100	16	1111111111010110
B/7	16	1111111111010101	16	1111111111010111
B/8	16	1111111111010110	16	1111111111011000
B/9	16	1111111111010111	16	1111111111011001
B/A	16	1111111111011000	16	1111111111011010
C/1	10	1111111010	9	111111010
C/2	16	1111111111011001	16	1111111111011011

续表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
C/3	16	1111111111011010	16	1111111111011100
C/4	16	1111111111011011	16	1111111111011101
C/5	16	1111111111011100	16	1111111111011110
C/6	16	11111111110U101	16	1111111111011111
C/7	16	1111111111011110	16	1111111111100000
C/8	16	1111111111011111	16	1111111111100001
C/9	16	1111111111100000	16	1111111111100010
C/A	16	1111111111100001	16	1111111111100011
D/1	11	11111111000	11	11111111001
D/2	16	1111111111100010	16	1111111111100100
D/3	16	1111111111100011	16	1111111111100101
D/4	16	1111111111100100	16	1111111111100110
D/5	16	1111111111100101	16	1111111111100111
D/6	16	1111111111100110	16	1111111111101000
D/7	16	1111111111100111	16	1111111111101001
D/8	16	1111111111101000	16	1111111111101010
D/9	16	1111111111101001	16	1111111111101011
D/A	16	1111111111101010	16	1111111111101100
E/1	16	1111111111101011	14	11111111100000
E/2	16	1111111111101100	16	1111111111101101
E/3	16	1111111111101101	16	1111111111101110
E/4	16	1111111111101110	16	1111111111101111
E/5	16	1111111111101111	16	1111111111110000
E/6	16	1111111111110000	16	1111111111110001
E/7	16	1111111111110001	16	1111111111110010
E/8	16	1111111111110010	16	1111111111110011
E/9	16	1111111111110011	16	1111111111110100
E/A	16	1111111111110100	16	1111111111110101
F/0	11	11111111001	10	1111111010
F/1	16	1111111111110101	15	111111111000011
F/2	16	1111111111110110	16	1111111111110100
F/3	16	1111111111110111	16	1111111111110111
F/4	16	1111111111111000	16	1111111111111000
F/5	16	1111111111111001	16	1111111111111001
F/6	16	1111111111111010	16	1111111111111010
F/7	16	1111111111111011	16	1111111111111011
F/8	16	1111111111111100	16	1111111111111100
F/9	16	1111111111111101	16	1111111111111101
F/A	16	1111111111111110	16	1111111111111110

9.7.3 渐进式DCT方式编码

基于DCT的顺序模式编码是对每一幅图像子块(8×8子块)按从左到右、从上到下的顺序一次扫描完成编码,而渐进式DCT方式编码模式是对每一幅图像子块的编码要经过若干次扫描才能完成。第一次只进行较粗糙的图像扫描压缩,并以相对于总的传输时间快得多的时间传输粗糙图像,重建质量较低的可识别图像。在随后的扫描中再对图像进行较细的压缩,且仅传递新增加的信息,重建一幅质量提高的图像,这样不断累进,直到获得满意的图像为止。累进的方式可采用频谱选择法或按位逼近法。

1. 频谱选择法

在每一次扫描中,只对DCT的64个变换系数中的某些频带的系数进行编码、传递,而其他频带的系数编码与传递在随后的扫描中进行,直到全部DCT系数处理完毕为止。

2. 按位逼近法

按DCT量化系数的有效位方向,即表示系数精度的位数方向分段渐进编码。第一次扫描只取最高有效位的n位编码和传递,然后再对其余位进行编码和传递。

9.7.4 JPEG 2000

JPEG 2000标准是JPEG工作组于2000年年底正式公布的JPEG最新静止图像压缩编码的国际标准,标准号为ISO/IEC 15444(ITU-TT.800)。

在JPEG 2000颁布之前,JPEG标准的基本系统已经在静止图像数据压缩方面得到了非常广泛的应用,在JPEG标准获得巨大成功的同时,该标准存在的一些缺点和不足也开始显现出来,主要表现在医学图像、多媒体、数字图书馆、Internet和移动网络等某些应用方面,尽管JPEG扩展系统致力于解决上述问题,但仅仅是在非常有限的范围之内发挥作用,并且扩展系统有时还因为知识产权和专利等方面的原因使应用受到限制。因此,JPEG工作组于1996年开始致力于研究一种新的静止图像压缩编码标准,这就是后来正式定名为JPEG 2000的国际标准。

为了能够用单一的压缩码流提供多种性能,满足更为广泛的应用,并考虑到JPEG标准已经广泛使用的现状,JPEG 2000主要由6个部分组成。第1部分为编码的核心部分,是公开并可免费使用的,它具有最小的复杂性却可以满足80%以上的图像压缩应用需求,其地位相当于JPEG标准的基本系统。第2~6部分则定义了压缩技术和文件格式的扩展部分。其中,第2部分为编码扩展,第3部分为Motion JPEG 2000(即MJP2),第4部分为一致性测试,第5部分为参考软件,第6部分是关于混合图像文件格式方面的内容和方法。

图9-13所示是采用了以上新技术的JPEG 2000的基本模块组成,其中包括预处理、DWT、量化、自适应算术编码以及码流组织等多个模块。

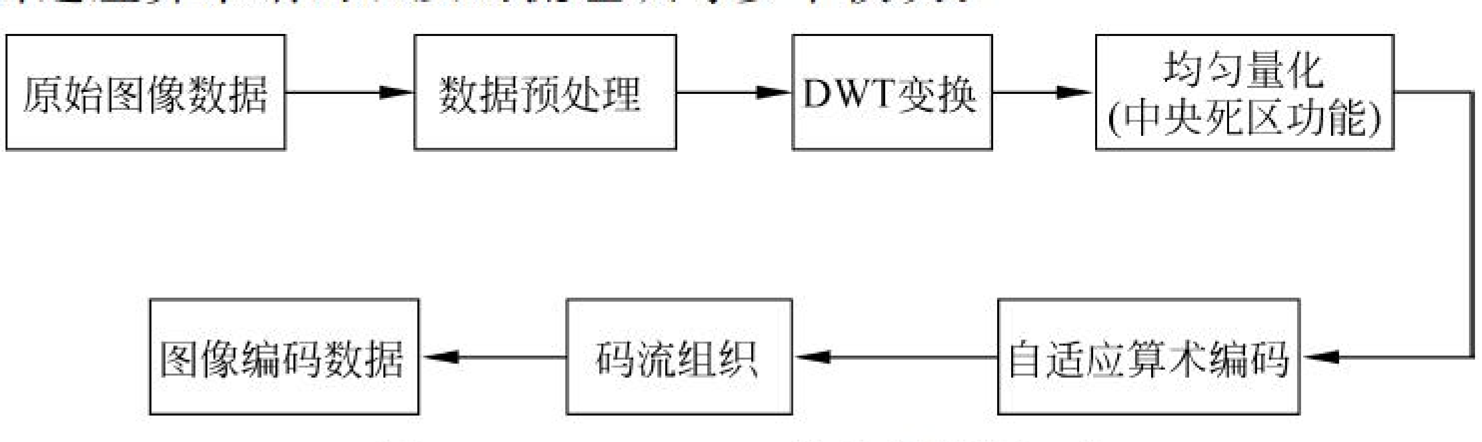


图9-13　JPEG 2000的基本模块组成

需要强调的是，JPEG 工作组在探索和研究静止图像压缩的新标准时提出，JPEG 2000 不仅应提供比 JPEG 基本系统更高的压缩性能，而且应具有一种关于图像新的描述方法，可以采用单一码流提供适应多种应用的性能。以能满足绝大多数图像压缩应用的 JPEG 2000 基本系统为例，JPEG 工作组主要在以下 9 个有代表性的方面提出了改进要求。

(1) 更高的图像数据压缩比。

(2) 同时支持有失真和无失真数据压缩。

(3) 很强的抗误码能力。

(4) 支持多分辨率。

(5) 对多重压缩/解压缩循环性能进行改进。

(6) 嵌入式码流(逐渐显示解码和 SNR 可分级)。

(7) 码流的随机存取和处理。

(8) 叠置(tiling)。

(9) 更灵活的文件格式。

为了满足新的性能标准和应用需求，JPEG 2000 采用了许多新的图像压缩编码技术，对包括压缩比、压缩算法和失真性等数据压缩技术的重要指标方面进行了重新考虑。

(1) JPEG 2000 系统中，以全帧离散小波变换(DWT)取代了 JPEG 基本系统基于图像子块的 DCT 变换，DWT 变换继承了小波变换的多分辨率性能，可以实现对图像的多分辨率表示，与人类的视觉特性非常吻合。此外，DWT 可以在大范围消除图像数据的相关性，更有效地集中图像能量，改善图像数据的分布，使数据压缩效率得到进一步提高。

(2) JPEG 2000 系统采用了一种具有中央“死区”功能的均匀量化器实现嵌入式编码方法。对于量化系数，各比特面采用基于上下文的自适应算术编码方式，这些由比特面提供的嵌入式码流同时也提供了 SNR 的可分级性；而且每个子带的比特面被限制在独立的矩形块中通过 3 次扫描完成编码，由此得到最佳的嵌入式码流、更高的抗误码能力、部分空间随机存取与处理能力，实现了非常灵活的码流语法。

(3) JPEG 2000 系统使用了整系数 DWT 滤波器，在单一码流中可以同时实现有失真和无失真图像数据压缩。

9.7.5 JPEG 编码实例

例 9-7：JPEG 编码举例

设一幅图像的 8×8 图像子块的亮度数据如图 9-14 所示，试对该子块按 JPEG 标准的基本系统进行编码。

97	94	90	84	78	76	67	51
96	93	89	83	77	75	66	50
95	92	88	81	76	74	64	49
93	90	87	80	74	72	63	47
91	88	84	78	72	70	61	45
89	86	82	75	70	68	58	43
86	83	80	73	67	66	56	40
84	81	77	70	65	63	53	38

图 9-14 原始图像 8×8 子块亮度数据

根据 JPEG 编码原理,基本系统编码过程的主要步骤如下:

(1) 对原始图像 8×8 子块的亮度数据进行 DCT 变换,结果如图 9-15 所示。

591	108	–21	24	–18	14	18	3
35	0	0	0	0	0	0	0
–5	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
–1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

图 9-15　原始图像 8×8 子块 DCT 系数表

(2) DCT 变换以后,用图 9-11(a)所示的亮度量化表对如图 9-15 所示的 DCT 系数矩阵进行量化,结果如图 9-16 所示。

37	10	–2	2	–1	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

图 9-16　DCT 系数量化结果

(3) 对量化结果表按规定顺序进行 Z 形扫描,并对图 9-16 中的一个 DC 及 5 个非零 AC 系数进行编码。

(4) DC 系数编码,Z 形扫描结果的第一个系数为 DC 系数。假设前一亮度数据块的 DC 系数为 17,则差分值为 20(37－17),查表 9-11 得 SSSS 为 3,其前缀码字为 110,尾码即为 20 的 5 位二进制原码 10100,从而 DC 系数的编码为 11010100。

(5) AC 系数编码

① 在图 9-16 中,按 Z 形顺序扫描,第 1 个非零 AC 系数为 10,该系数之前的 0 的连续个数为 0,即 NNNN＝0,根据系数 10,查表 9-12 得,SSSS＝4,因此,NNNN/SSSS＝0/4,查表 9-13 得霍夫曼编码为 1011,加上 10 的二进制编码为 1010,由此可得 AC 系数 10 的编码为 10111010。

② 第 2 个非零 AC 系数为 3,该系数之前的 0 的连续个数为 0,即 NNNN＝0,根据系数 3,查表 9-12 得,SSSS＝2,因此,NNNN/SSSS＝0/2,查表 9-13 得霍夫曼编码为 01,加上 3 的二进制编码为 11,由此可得 AC 系数 3 的编码为 0111。

③ 第 3 个非零 AC 系数为－2,该系数之前的 0 的连续个数为 2,即 NNNN＝2,根据系数－2,查表 9-12 得,SSSS＝2,因此,NNNN/SSSS＝2/2,查表 9-13 得霍夫曼编码为

11111001,加上−2 的绝对值之二进制编码为 10,其反码为 01,由此可得 AC 系数−2 的编码为 1111100101。

④ 第 4 个非零 AC 系数为 2,该系数之前的 0 的连续个数为 0,即 NNNN=0,根据系数 2,查表 9-12 得,SSSS=2,因此,NNNN/SSSS=0/2,查表 9-13 得霍夫曼编码为 01,加上 2 的二进制编码为 10,由此可得 AC 系数 2 的编码为 0110。

⑤ 第 5 个非零 AC 系数为−1,该系数之前的 0 的连续个数为 7,即 NNNN=7,根据系数−1,查表 9-12 得,SSSS=1,因此,NNNN/SSSS=7/1,查表 9-13 得霍夫曼编码为 1111010,加上−1 的绝对值之二进制编码为 1,其反码为 0,由此可得 AC 系数 1 的编码为 11110100。

⑥ 由于其他 AC 系数全为 0,直接以结束块“EOB(0/0)”结束 8×8 子块的编码,查表 9-13 得编码为 1010。因此,可得该 8×8 子块的全部编码为

11010100101111010011111111001010110111101010010

共有 46 位编码,而原始图像子块需要 8×8×8=512 位,由此可得压缩比为 11∶1。

9.8 MPEG 标准

MPEG 是 ISO 运动图像专家小组的英文缩写,该组织专门致力于对运动图像压缩编码技术的标准化与应用研究。该小组于 1991 年年底提出了用于数字存储媒介、速率约 1.5MB/s 的运动图像与伴音的压缩、解压缩及编码的国际标准,并于 1992 年正式通过,通常被称为 MPEG-1 标准,即 ISO/IEC11172。

目前标准不是一个单一的标准,而是一个用于全运动视频和相关音频压缩的标准系列,包括 MPEG-1、MPEG-2、MPEG-4、MPEG-7、MPEG-21 等多个标准(MPEG-3 不常用),如表 9-14 所示,每一个 MPEG 标准都有其特定的应用范围。其中,MPEG-1、MPEG-2、MPEG-4 标准的应用范围最广。MPEG 标准的视频压缩编码技术主要利用了具有运动补偿的帧间压缩编码技术以减小时间冗余度,利用 DCT 技术以减小图像的空间冗余度,利用熵编码则在信息表示方面减小了统计冗余度。这几种技术的综合运用,大大增强了压缩性能。

表 9-14 MPEG 标准

标准号	应用范围	数据率
MPEG-1	320×240 广播级全运动图像	1.5Mb/s 标准 TV 质量输出
MPEG-2	720×480 广播级全运动图像	4~10Mb/sVCR 质量输出
MPEG-4	适用于窄带视频电话和具有低分辨率小幅图像	9~40kb/s 视频电话图像输出

9.8.1 MPEG-1

MPEG-1 主要包括系统、视频、音频、一致性、参考软件 5 部分。其目的是满足对各种存储媒体上压缩视频统一表示格式的需要,可用于 625 线和 525 线电视系统,对传输速率约 1.5Mb/s 的存储媒体提供连续、活动图像的编码表示(如 VCD、DAT 以及计算机磁盘等),但因采用的方法是通用的,因此可用于更广泛的范围。

1. 图像格式与约束参数

MPEG-1 处理的对象是逐行扫描的图像，对于交织的图像源，编码前必须先转换为非交织格式；输入信号（视频信号）必须是数字化的 YC_bC_r 信号，数据经过预处理和格式转换后选择一个合适的窗口、分辨率和输入格式，要求色差信号和亮度信号在垂直与水平方向按 2∶1 进行抽样。

由于 MPEG-1 可表示的比特流的特性范围很大，因此定义了一个如表 9-15 所示的约束参数编码参数子集，并用比特流中的一个标志位进行标识。当码率为 1～1.5Mb/s 时，图像速率的合理选择是每秒 24、25 或 30 幅，水平分辨率在 250～400 像素，垂直分辨率在 200～300 线。

表 9-15　约束参数表

项　　目	指　　标
图像宽度	≤768 个像素
图像高度	≤576 线
图像范围	≤396 个宏块
像素速率	≤396×25 宏块/秒

2. 算法及编解码

MPEG 编码技术的选择是基于高质量的连续活动图像、高压缩比以及对编码比特流的随机操作需求之间的平衡。运动图像的压缩包括帧内压缩与帧间压缩两个主要方面。帧内压缩是删除基于空间的数据冗余，帧间压缩则是删除基于帧与帧之间的时间冗余。MPEG 标准在空间域的压缩，吸收了 JPEG 的一些要点，将每一帧作为独立的图像获取，其压缩步骤与 JPEG 标准基本一致。而运动图像相邻帧之间（视频图像信号）具有非常大的相关性，即存在很大的时间冗余，多数情况下，从一帧到下一帧变化的只是一小部分，而画面中的绝大部分相同。运动图像时间冗余的删除是通过帧间编码完成的。

帧间编码的基本思想是只对运动图像从一帧到下一帧的变化部分进行存储，而并非存储全部图像数据，因而可以极大地减少运动图像数据的存储量，达到帧间压缩的目的。如表 9-16 所示，将帧序列划分成 I 帧、P 帧、B 帧 3 种类型，使用参照帧及运动补偿技术来实现帧间编码。

表 9-16　帧序列的分类

帧 类 型	意　　义
I 帧	在解码时，无须参照任何其他帧的帧图像称为 I 帧，或称为内编码帧，它是利用自身的相关性进行帧内压缩编码，它给编码序列的解码的起始提供操作点，满足随机操作的要求，但仅能获得中等的编码压缩比
P 帧	在帧编码时，仅使用最近的前一帧（I 帧或 P 帧）作为参照帧时，该帧称为 P 帧，或称为预测帧；P 帧利用过去的 I 帧或 P 帧进行运动补偿预测，可得到更有效的编码
B 帧	在帧编码时，要使用前后帧作为参照帧时，该帧称为 B 帧，或称为双向预测帧；B 帧能提供最大限度的压缩，它需要过去和将来的参考图（I 帧或 P 帧）进行运动补偿，但它本身不能用作预测参考图

提高帧间压缩的有效方法是通过采用运动补偿技术，在对 P 帧或 B 帧进行编码时，以宏块（一个宏块一般可定义为 16×16 数据块）为基本编码单位，运动补偿技术可以消除 P 帧和 B 帧在时间上的冗余。

MPEG-1 没有规定编码过程，而仅规定了比特流的语法和语义以及解码器中的信号处理。在有 B 帧时，要有两个帧存储器分别存储过去和将来的两个参考帧，以便进行双向运动补偿预测。编码器必须在图像质量、编码速率以及编码效率之间进行综合考虑，选择合适的编码工作模式和控制参数。此外，标准在一些具体模块的实现上具有开放性，可以自行选用诸如运动矢量估计算法、图像的刷新机制、编码控制机制等。

MPEG-1 的解码无须运动估计，只要根据接收到的码流的语义进行相应的处理。当一幅图像的所有宏块都处理完毕，则整个图像被重建。如果编码序列包括 B 帧，则解码后还应按显示顺序重新排序后才能进行显示。

3. 编码视频流的层次结构和组成

MPEG-1 规定了编码视频比特流的语法。该语法分为 6 层，如表 9-17 所示。最高层为序列层，最低层为块层，每层支持一个信号处理或一个系统功能，并且设定了必要的参数。

表 9-17 MPEG-1 语法表

语义层	功能
序列层	随机存取单元：上下文
图像组层	随机存取单元：视频
图像层	初始编码单元
宏块条层	重同步单元
宏块层	运动补偿单元
块层	DCT 单元

9.8.2 MPEG-2

MPEG-2 标准是 MPEG 工作组制定的第二个国际标准，即通用的活动图像及其伴音的编码标准 ISO/IEC 13818，于 1994 年制定，目标是高级工业标准的图像质量以及更高的传输率。它主要包括以下几个部分：

(1) 系统。

(2) 视频。

(3) 音频。

(4) 一致性。

(5) 参考软件。

(6) 数字存储媒体的命令与控制。

(7) 高级音频编码。

(8) 10bit 视频编码。

(9) 实时接口。

对于视频部分，作为一个通用的编码标准，MPEG-2 比 MPEG-1 具有更广泛的适应范围，能提供的传输率在 3～10Mb/s 间，其在 NTSC 制式下的分辨率可达 720×486，MPEG-2 也可提供并能够提供广播级的视像和 CD 级的音质。MPEG-2 可适应于标准数字电视、高清电视，同时也包括 MPEG-1 的工作范围，音频编码可提供左右中及两个环绕声道，以及一个加重低音声道和多达 7 个伴音声道。由于 MPEG-2 在设计时的巧妙处理，使得大多数 MPEG-2 解码器也可播放 MPEG-1 格式的数据，如 VCD。

MPEG-2 的解码器可以对 MPEG-1 码流进行解码,并在其基础上增加了一些新的编码技术,MPEG-2 规定了 4 种图像的运动预测和补偿方式,引入了空间可分级性、时间可分级性以及信噪比 SNR 可分级性 3 种编码的可分级性。因此,甚至有人认为,MPEG-1 已经成为 MPEG-2 的一个子集。

同时,由于 MPEG-2 的出色性能表现,已能适用于 HDTV,使得原打算为 HDTV 设计的 MPEG-3,还未完全推出就被放弃了。除了作为 DVD 的指定标准外,MPEG-2 还可用于为广播、有线电视网、电缆网络以及卫星直播提供广播级的数字视频。

MPEG-2 的另一特点是,其可提供一个较广的范围改变压缩比,以适应不同画面质量、存储容量,以及带宽的要求。

9.8.3 MPEG-4

MPEG-4 即 ISO/IEC 14496,该标准是一个可适应各种多媒体应用的"音频视觉对象编码"标准,MPEG-4 实现了从矩形帧到 VOP 的转变以及基于像素的传统编码向基于对象和内容的现代编码的转变,这正体现了传统视频编码与新一代视频编码的有机统一。基于内容的交互性是 MPEG-4 的核心思想,它利用很窄的带宽,通过帧重建技术,压缩和传输数据,达到以最少的数据获得最佳的图像质量,这对于视频编码技术的发展方向及广泛应用都具有特别重要的意义。

MPEG-4 具有版本 1 和版本 2 两个版本。版本 1 于 1998 年 10 月通过,1999 年 2 月公布,可以分为如下 7 个部分。

(1) 系统。

(2) 视觉信息。

(3) 音频。

(4) 一致性。

(5) 参考软件。

(6) 多媒体传送集成框架。

(7) MPEG-4 工具(视频)优化软件。

版本 2 于 1999 年 12 月通过,MPEG-4 标准包含系统、视频、音频、一致性和 3D 图形一致性检查等 27 个部分。MPEG-4 规定了各种音频视觉对象的编码,除了包括自然的音频视频对象以外,还包括图像、文字和 3D 图形以及合成话音和音乐等。MPEG-4 的编码可以适应多种媒体的应用需求,它通过描述场景对象的空间位置和时间关系等,来建立多媒体场景,并将它与编码的对象一起传输。由于对各个对象进行独立的编码,可以达到很高的压缩效率,同时也为在接收端根据需要对内容进行操作提供了可能,可适应多媒体的人机交互需求。MPEG-4 作为新一代多媒体数据压缩编码的典型代表,其领先思想是首次提出了基于内容、基于对象的压缩编码思想。它要求对自然或合成视听对象做更多分析和理解,一定程度上代表了现代数据压缩编码技术的发展方向。

习题

1. 数字图像的信息冗余主要包括哪几方面?
2. 什么是图像信息的视觉冗余?

3. 图像编码的效率指标主要包括哪些?

4. 图像数据压缩的评价指标有哪些? 这些指标是如何定义的?

5. 试简述什么是变长最佳编码定理。

6. 试简述霍夫曼编码算法的基本原理和步骤。

7. 试分析霍夫曼编码有哪些特点。

8. 设一幅灰度级为5的图像中,各灰度级分别用 S_0、S_1、S_2、S_3、S_4 表示,对应的概率分别为0.375、0.175、0.175、0.15、0.125,试采用霍夫曼编码方法对其进行编码。

9. 编写MATLAB程序,实现图像的霍夫曼编码。

10. 香农—范诺编码的理论基础是什么? 简述该编码的主要步骤。

11. RLE编码的原理是什么? 怎样实现RLE编码?

12. 算术编码具有哪些特点?

13. 试简述算术编码的原理和主要步骤。

14. 设图像信源编码可用 a、b、c、d 共4个符号表示,若图像信源字符集为{$bcabd$},信源字符出现的概率分别如表9-18所示,采用算术编码对该图像字符集进行编码。

表9-18 图像信源字符出现概率

信源字符	a	b	c	d
出现概率	0.2	0.4	0.2	0.2

15. 什么是位平面编码? 灰度图像位平面分解中,高位面和低位面有何特点?

16. 预测编码主要包括哪几种类型?

17. 试简述图像变换编码的基本原理。

18. 试简述静止图像压缩编码的技术标准JPEG由几部分组成。

19. JPEG标准的DCT变换系数量化具有哪些作用?

20. DCT系数量化后,为什么需进行Z形扫描?

21. 试简述DCT系数编码的主要步骤。

22. 与JPEG基本系统相比,JPEG 2000采用了哪些新的图像压缩编码技术?

23. 若一幅图像的8×8图像子块的亮度数据如图9-17所示,试对其按JPEG基本系统进行编码。

128	121	109	78	72	64	56	62
139	126	102	74	75	66	68	87
109	102	98	89	76	68	89	103
97	117	117	78	74	94	103	79
164	149	88	87	99	91	74	68
147	94	90	102	84	72	82	102
95	92	116	119	114	122	137	150
111	112	140	150	157	163	161	157

图9-17 原始图像8×8子块亮度数据表

第 10 章 数学形态学及应用

CHAPTER 10

数学形态学(mathematical morphology)理论诞生于 1964 年,法国巴黎矿业学院博士生赛拉(J. Serra)和导师马瑟荣(G. Matheron)在从事铁矿矿山的定量岩石学分析及预测其开采价值时,提出了击中/击不中变换、开闭运算等概念和理论,从此开启了形态学的应用研究。

数学形态学的基本思想是用具有一定形态的结构元素去量度和提取数字图像中的对应形状,以实现对图像进行分析和识别的目的。形态学分析是一门建立在集合论基础上的学科,它是几何形态分析和描述的有力工具,是一种新的图像处理与分析方法。近年来,形态学分析在数字图像和机器视觉领域得到了广泛应用,其基本思想是用一定形态的结构元素去度量和检测被度量图像中的对应形状,以达到对图像进行分析和识别的目的。

数学形态学的基本思想也适应于图像处理的很多方面,包括图像增强、边缘检测、图像分割、特征提取、图像复原、文字识别、医学图像处理、图像压缩、材料科学以及机器视觉等众多领域。在工农业生产中,视觉检测、零部件检测、产品质量检测、食品安全检测、生物医学图像分析和纹理分析等方面取得了非常成功的应用,创造了较好的经济效益和社会效益。从某种意义上讲,数学形态学已经构成了一种新型的数字图像分析方法和理论,并已成为图像工程技术人员进行图像分析的得力助手。本章先介绍数学形态学的基本概念、定义和理论,然后介绍二值数学形态学的基本理论、方法与算法,再介绍灰度形态学,最后以二值形态学为例介绍数学形态学在图像处理中的应用。

10.1 数学形态学基础

10.1.1 形态学简介

赛拉和马瑟荣在进行岩相学的定量分析时第一次引入了形态学的数学表达式及理论,建立了颗粒分析方法。他们师徒二人在提出击中/击不中变换和开闭运算之后,又提出了布尔模型及纹理分析器原型等概念和理论,奠定了数学形态学的基础理论和工程应用基础。

数学形态学的主要方法是采用具有一定形态的“结构元素”去量度和提取图像中的对应形状,从而实现对图像进行分析和识别。形态学的基础主要基于集合论,因此它具有坚实而完备的数学基础,数学形态学完备的理论基础为它用于图像分析和处理、形态滤波器的特性分析和系统设计提供了依据。数学形态学用于图像分析与处理可以简化图像数据、保持它

们基本的形状特性，并除去不相干的结构。数学形态学的计算方法具有天然的并行实现结构，实现了形态学分析和处理算法的并行计算，提高了图像分析和处理的速度。

在数学形态学中，用集合来描述图像目标以及图像各部分之间的关系，说明目标的结构特点。数学形态学由一组形态学的代数运算子组成，其基本运算如下：

(1) 膨胀与腐蚀。

(2) 开运算与闭运算。

(3) 击中与击不中。

(4) 细化与粗化。

这些基本运算子在二值图像和灰度图像中各有特点，基于这些基本运算还可推导和组合构成其他数学形态学的实用算法，用它们可以完成包括图像分割、特征抽取、边缘检测、图像滤波等在内的图像形状和结构的分析与处理。

数学形态学用集合的概念对图像目标进行描述，描述图像各部分之间的关系，分析目标的结构特点。在分析和考察图像时，要设计一种收集图像信息的“探针”，称为结构元素。结构元素通常比待分析和处理的图像简单，尺寸也要小一些，如圆形、正方形、线段的集合可作为结构元素。当被处理的图像是二值图像时，结构元素也采用二值图像；当被处理的图像是灰度图像时，则采用灰度图像作为结构元素。处理图像时，通过在图像中不断移动结构元素，分析和考察图像中各部分之间的关系，提取有用的信息进行结构分析和描述。使用不同的结构元素和形态学算子可以获得目标的大小、形状、连通性和方向等重要信息。数学形态学基于探测的思想，与人类的FOA(focus of attention)的视觉特点有类似之处。作为探针的结构元素，其形态、大小、灰度和色度信息携带有目标的知识信息，因此可以分析和探测图像的结构特点。

数学形态学的基本思想及方法可适用于图像处理的很多方面，如击中/击不中变换可以用于目标识别，流域的概念可以用于图像分割，腐蚀和开运算可以用于骨架抽取，形态学滤波器可以用于颗粒分析等。可以说，目前还没有一种方法能像数学形态学一样，既有严密的理论基础，又具有简洁、朴素、统一的表达方法和思想，而且具有广泛的应用价值。因此，有人说数学形态学理论严密、完备，概念清晰，应用简单。

数学形态学是建立在严密数学理论基础上的应用学科，其方法对图像处理的理论和工程应用产生了重要的影响。不可否认，数学形态学已成为一种新的图像处理方法，并已应用到数字图像分析、处理及一些其他学科上。数学形态学方法在计算机文字识别、显微图像分析(定量金相分析、颗粒分析)、医学图像处理(细胞检测、CT/MRI图像处理)、图像编码、工业检测(如食品安全检测、印刷电路检测、产品质量检测)、材料科学、指纹检测、地理信息、合成音乐、机器视觉等工程方面的应用已趋于成熟，并在不断开拓新的应用领域。

10.1.2 术语和定义

数学形态学的理论建立在集合论的基础之上，因此在介绍数学形态学的算法之前，先介绍一些集合论和数学形态学的符号和术语。

1. 元素与集合

(1) 集合：将一些可区别的客体，按其共同特性进行汇集，具有共同特性的这些客体的全体成员称为集合。例如，图像中灰度级相同的区域就可构成一个集合。集合通常用大写

字母 $A,B,C,\cdots$表示。如果某种客体不存在，就称这种客体的全体是空集，记为$\varnothing$。

(2)元素：组成集合的各个客体，称为该集合的元素，又称为集合的成员。如一幅图像中物体上的任一像素都可以是元素。常用小写字母 $a,b,c,\cdots$表示元素。任何客体都不是空集$\varnothing$的元素。若 $a\in A$，表示 a 是集合 A 的元素。

(3) 子集：集合 A 包含集合 B 的充要条件是集合 B 的每个元素都是集合 A 的元素，也可以称为集合 B 包含于集合 A。记为 $B\subseteq A$，即 B 包含于 A，或 A 包含 B。又称 B 是 A 的子集。若集合 A 与 B 相等，必有 $B\subseteq A$，同时 $A\subseteq B$。

在基于数学形态学的数字图像处理与运算中，一幅图像可称为一个集合。对于二值图像而言，习惯上认为取值为 1 的点对应于景物中心，用阴影表示，而取值为 0 的点组成背景，用白色表示，这类图像的集合是直接表示的。考虑所有值为 1 的点的集合为 A，则 A 与图像是一一对应的。对于一幅图像 A，如果点 a 在 A 的区域之内，那么就说 a 是 A 的元素，记为 $a\in A$，若 b 不在 A 的区域之内，则 b 不是 A 的元素，记作 $b\notin A$，如图 10-1 所示。

如图 10-2 所示，对于两幅图像 A 和 B，若图像 B 中的每一个点 b，如果 $b\in A$ 都成立，则称 B 包含于 A，记作 $B\subseteq A$。如果图像 A 中至少存在一个点 a，即 $a\in A$，且 $a\notin B$，则称 B 是 A 的真子集，记作 $B\subset A$。

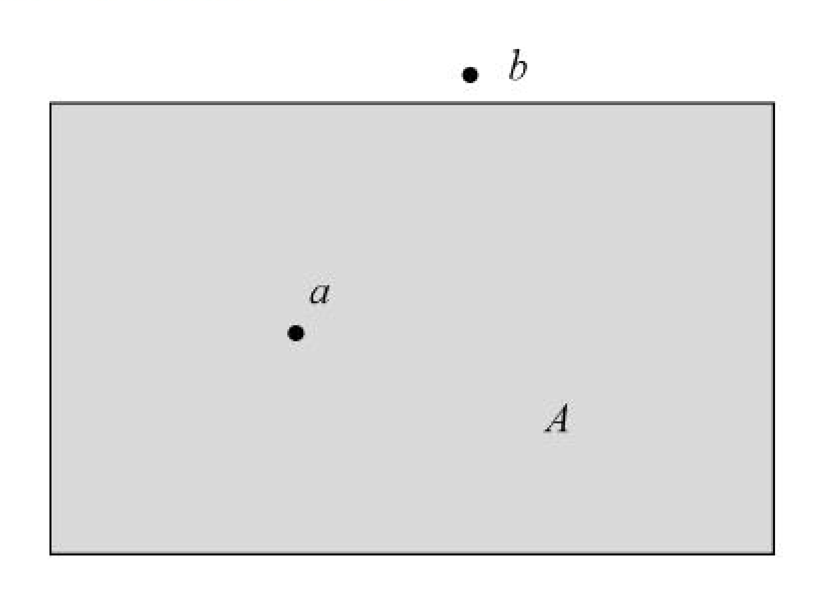


图 10-1　元素与集合间的关系

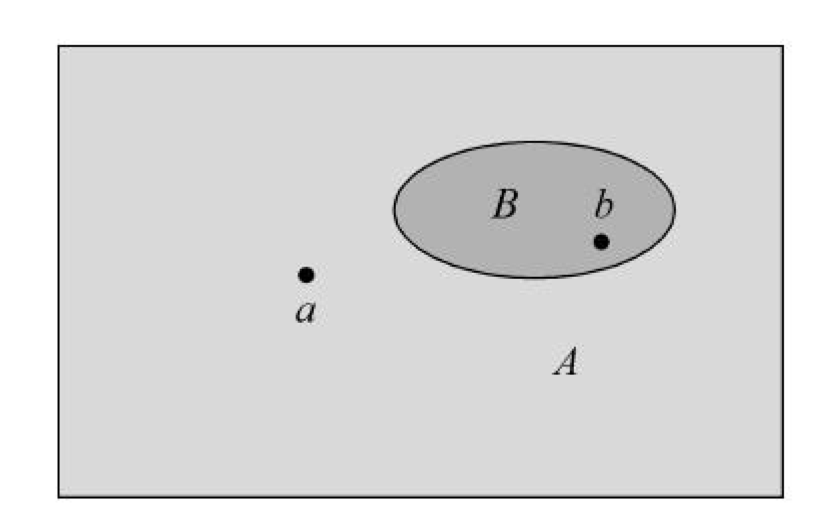


图 10-2　集合之间的关系

根据集合的定义可知，对于图像 A 和 B：

(1) 如果 $B\subset A$，那么必有 $B\subseteq A$；

(2) $A\subseteq A$ 恒成立。

2. 交集、并集和补集

交集、并集和补集是集合论中的基本运算。

1) 交集

两个图像集合 A 和 B 的公共点组成的集合称为两个集合的交集，记为 $A\cap B$，即

$$A \cap B = \{a \mid a \in A \text{ 且 } a \in B\}$$

2) 并集

两个集合 A 和 B 的所有元素组成的集合称为两个集合的并集，记为 $A\cup B$，即

$$A \cup B = \{a \mid a \in A \text{ 或 } a \in B\}$$

3) 补集

对一幅图像 A，在图像 A 区域以外的所有点构成的集合称为 A 的补集，记为 A^c，即

$$A^c = \{a \mid a \notin A\}$$

交集、并集和补集运算的几何图解分别如图 10-3(a)、(b)、(c)所示。

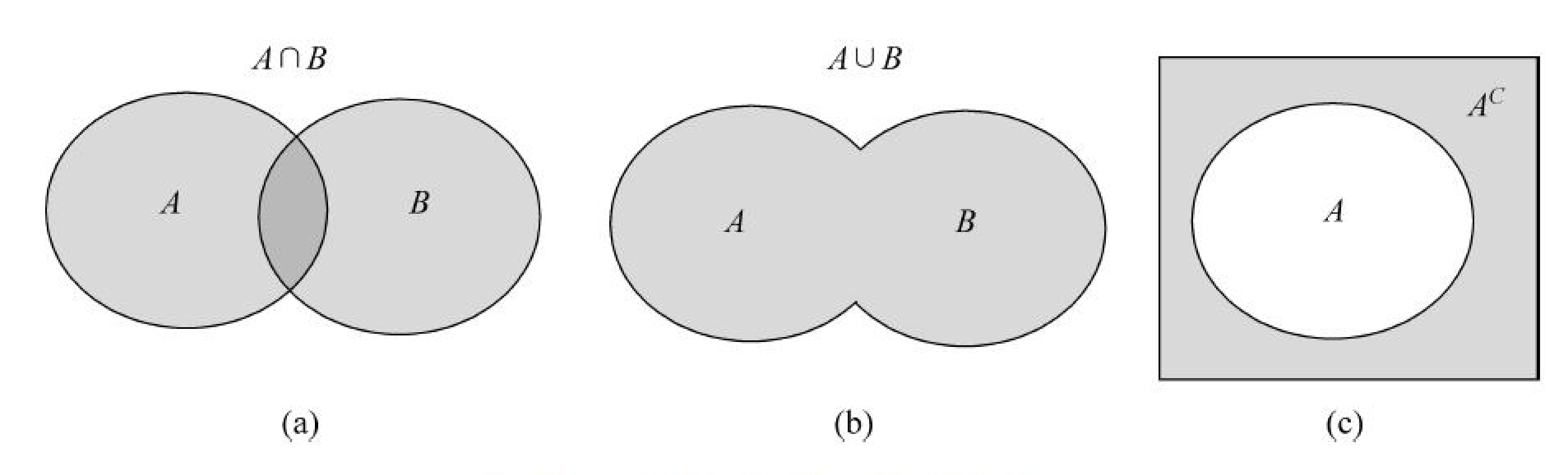


图 10-3　集合的交集、并集和补集运算

3. 击中与击不中

1）击中(hit)

如图 10-4 所示，设有两幅图像 A 和 B，如果 $A\cap B\neq\varnothing$，则称 B 击中 A，记为 $B\uparrow A$。

2）击不中(miss)

对于如图 10-5 所示的两幅图像 A 和 B，如果 $A\cap B=\varnothing$，则称 B 击不中 A。

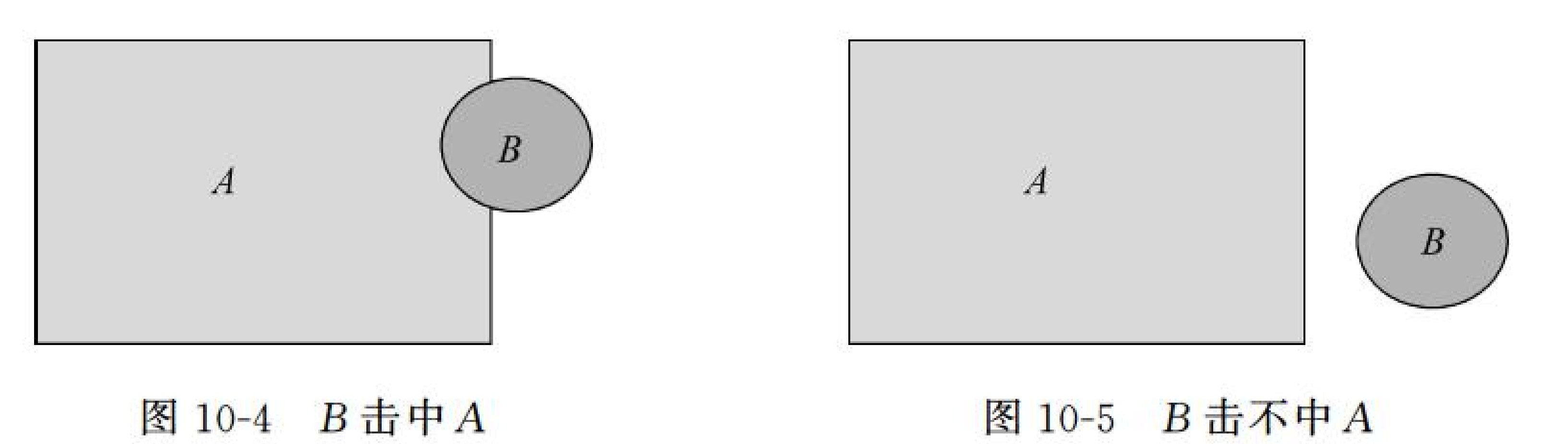
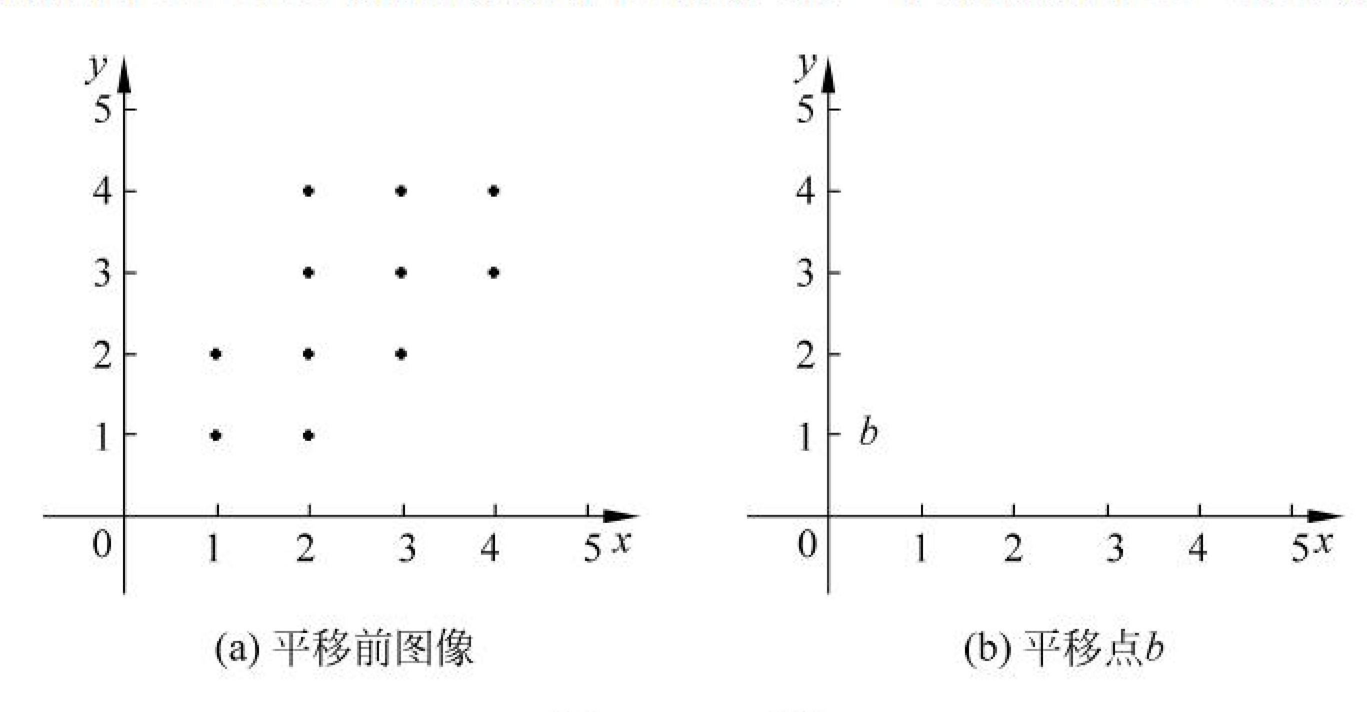


图 10-4　B 击中 A　　　图 10-5　B 击不中 A

4. 平移和映射

1）平移

设 A 是一幅如图 10-6(a)所示的数字图像，b 是一个点，如图 10-6(b)所示。



(a) 平移前图像　　(b) 平移点b

图 10-6　平移

定义 A 被 b 平移后的结果可表示为 $A+b=\{a+b|a\in A\}$，即取出 A 中的每一个点 a 的坐标值，将其与点 b 的坐标求和，得到一个新的点的坐标 $a+b$，这些新点所构成的图像就是 A 被 b 平移的结果，记为 $A+b$，如图 10-7 所示。

2）映射

映射又称为映像，还可以称为反射。A 的映射结果可表示为 $A^v=\{a|-a\in A\}$，即将 A 中的每一个点取反以后所得的新图像称为映射，如图 10-8 所示。

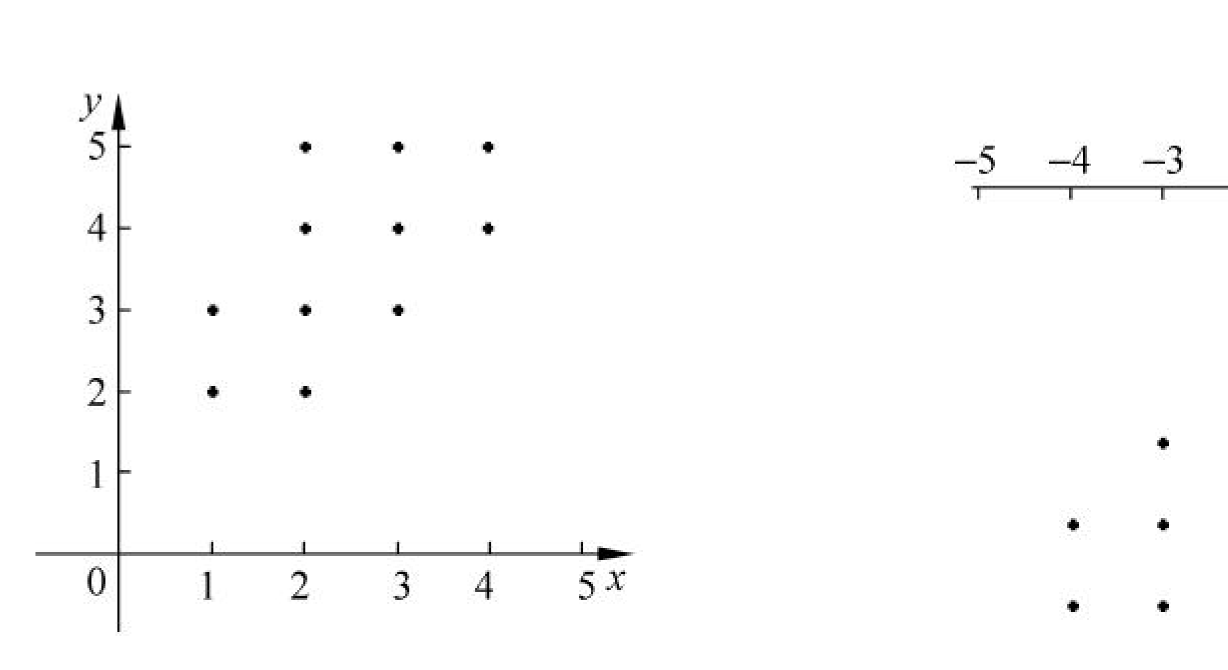


图 10-7 平移

图 10-8 映射

5. 结构元素

结构元素是数学形态学中一个非常最重要而又基础的概念。习惯上,数学形态学方法称被处理的图像为目标图像,一般用大写字母表示。为了确定目标图像的结构,必须逐一考察图像各部分之间的关系,并进行检验,最后得到一个各部分之间关系的集合。在考察目标图像各部分之间的关系时,需要设计一种收集信息的"探针",称为"结构元素"。结构元素具有一定的几何形状,圆形、正方形、十字形、有向线段的集合等都可以作为结构元素,分析时在目标图像中移动结构元素便可以考察图像中各部分之间的关系,从而提取有用的特征信息,实现对图像的分析和描述。

结构元素在形态学运算中的作用类似于在信号处理时的"滤波窗口"或"参考模板"的作用。对于每一个结构元素,需要指定一个原点,它是结构元素参与形态学运算的参考点,该原点可以包含在结构元素中,也可以不包含在结构元素中,但两者运算的结果可能不同。

结构元素同样用大写英文字母表示。通过不断移动结构元素,就可以研究图像中各部分之间的关系。一般结构元素的尺寸应明显小于目标图像的尺寸,结构元素的选取会直接影响形态运算的效果,因此,应根据具体情况确定结构元素。结构元素的选取一般应遵循以下原则:

(1) 结构元素必须在几何上比原图像简单并有界,尺寸应小于被分析的图像,当选择性质相同或相似的结构元素时,以选取图像某些特征的极限情况为宜。

(2) 结构元素的形状最好具有某种凸性,如圆形、十字架形、方形等。对非凸性子集,由于连接两点的线段有些部分位于集合的外面,即在其补集上,因此,采用非凸性子集作为结构元素难以提取更多的有用信息,在分析和讨论时一般采用圆形、十字架形、正方形等作为结构元素。

10.2 二值形态学

二值形态学的运算对象是集合。设 A 为一幅图像,S 为结构元素,实际上,结构元素自身也是一个图像(集合)。数学形态学运算是指用 S 对 A 进行运算或操作,对每个结构元素可以指定一个原点作为参考点。

10.2.1 腐蚀

腐蚀是最基本的一种数学形态学运算。腐蚀运算在数学形态学图像处理中的作用是可以消除物体的边界点。对一个给定的目标图像 X 和一个结构元素 S，将 S 在图像 X 上移动，运动到每一个当前位置 x，$S+x$ 只有如下三种可能的状态，如图 10-9 所示。

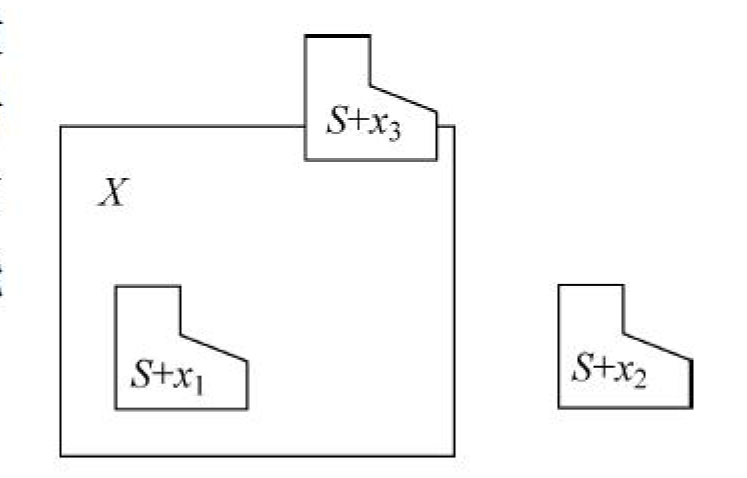


图 10-9　$S+x$ 的三种状态

(1) $S+x\subseteq X$。

(2) $S+x\subseteq X^{C}$。

(3) $S+x\cap X$ 与 $S+x\cap X^{C}$ 均不为空集。

情况(1)说明 $S+x$ 与 X 相关最大，情况(2)表明 $S+x$ 与 X 完全不相关，情况(3)表明 $S+x$ 与 X 存在部分相关。因而满足上述三条的全体 x 点的集合构成结构元素与图像 X 的最大相关点集，这个点集称为 S 对 X 的腐蚀(简称腐蚀，有时也称 X 用 S 进行腐蚀)，记为 $X\Theta S$。

腐蚀也可用集合来进行定义，即

$$X\Theta S=\{x \mid S+x\subseteq X\} \tag{10-1}$$

式(10-1)表明，X 用 S 腐蚀的结果是所有使 S 平移 x 后仍在 X 之中的全体 x 的集合。也就是说，用 S 来腐蚀 X 得到的集合是 S 完全包括在 X 中时 S 的原点位置的集合。

腐蚀运算可以消除物体的边界点。例如，若结构元素选取 3×3 的矩形块，对图像进行腐蚀运算将使原图像的边界沿周边减少一个像素。选取适当大小的结构元素，腐蚀运算可以将图像内小于结构元素的物体，如毛刺、小凸起等消除，通过取不同大小的结构元素，就可以在原图像中去掉不同大小的物体。因此，若两个物体之间有小的连通区域，当结构元素大小适当时，腐蚀运算可以将两个物体分开。

为加深对腐蚀运算的理解，接下来分析两个腐蚀运算实例。

例 10-1：腐蚀运算示例 1

如图 10-10 所示为腐蚀运算示例。如图 10-10(a)为原始图像(集合)A，图 10-10(b)为结构元素，图 10-10(c)为腐蚀运算结果。

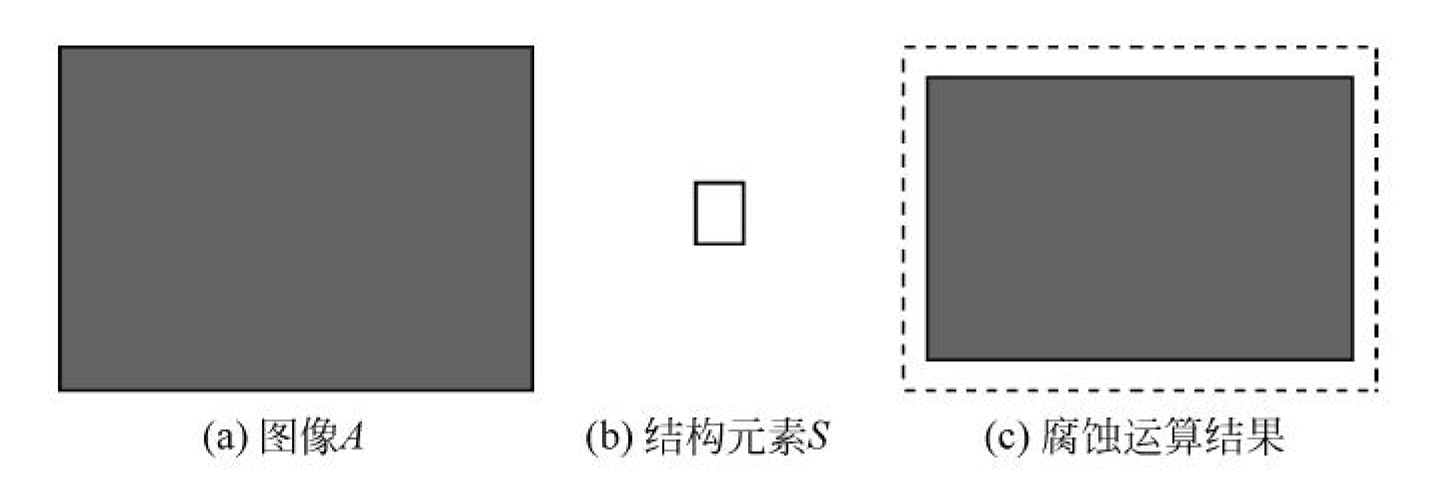


图 10-10　腐蚀运算

例 10-2：腐蚀运算示例 2

如图 10-11 为腐蚀运算的一个典型示例。图 10-11(a)中的阴影部分为集合 X，图 10-11(b)中的阴影部分为结构元素 S，而图 10-11(c)中的深黑色部分即 $X\Theta S$ 运算的结果。由图 10-11(c)可见，腐蚀运算使图像收缩变小了。

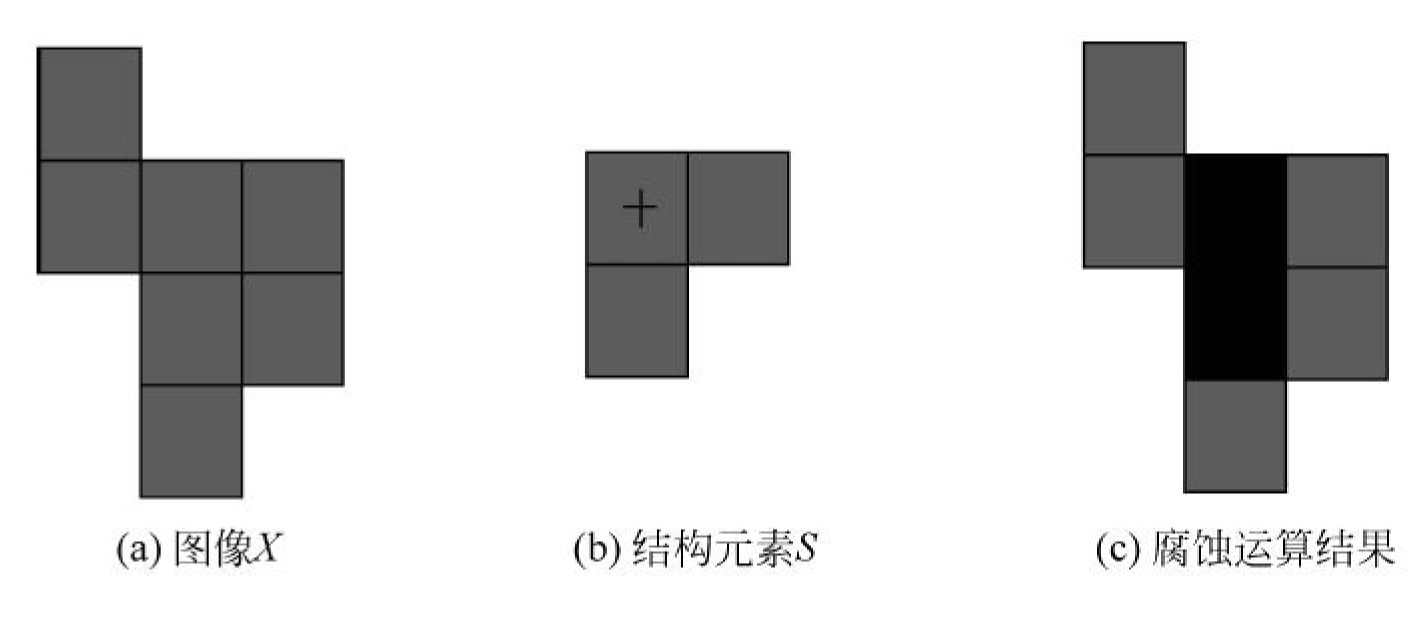


图 10-11　腐蚀示意图

通过观察上述两个腐蚀运算的例题，对腐蚀运算的作用有了进一步的理解。根据腐蚀运算的定义可以得知，若 S 包含了原点，即 $O \in S$，则 $X\Theta S$ 将是 X 的一个收缩，即 $X\Theta S \subseteq X$（当 $O \in S$ 时）；若 S 不包含原点，那么 $X\Theta S \subseteq X$ 并不一定成立。若结构元素 S 关于原点 O 是对称的，那么 $S = S^V$。

由此可得 $X\Theta S = X\Theta S^V$，但是，若 S 关于原点 O 不对称，则 X 被 S 腐蚀的结果与 X 被 S^V 腐蚀的结果不相同。根据腐蚀运算的定义可以直接给出腐蚀算法，腐蚀运算还有如下所示的另一种表达式：

$$X\Theta S = \bigcup \{S + X \mid - s \in S\} \qquad (10\text{-}2)$$

式(10-2)可从定义式(10-1)导出，它将腐蚀表示为图像平移的交，这在某些并行处理环境中比较具有应用价值。

10.2.2　膨胀

腐蚀可以视为将图像 X 中每一与结构元素 S 全等的子集 $S + x$ 收缩为点 x。反之，也可以将 X 中的每一个点 x 扩大为 $S + x$，这就是膨胀运算，记为 $X \oplus S$。膨胀运算定义为

$$X \oplus S = \{x \mid S + x \cup x \neq \phi\} \qquad (10\text{-}3)$$

例 10-3：膨胀运算示例

仍以图 10-10(a)所示图像(集合)A 和图 10-10(b)所示的结构元素 S 为例进行膨胀运算，即用结构元素 S 对 A 进行膨胀，其结果如图 10-12 所示(图中的虚线区域为原始图像 A)。

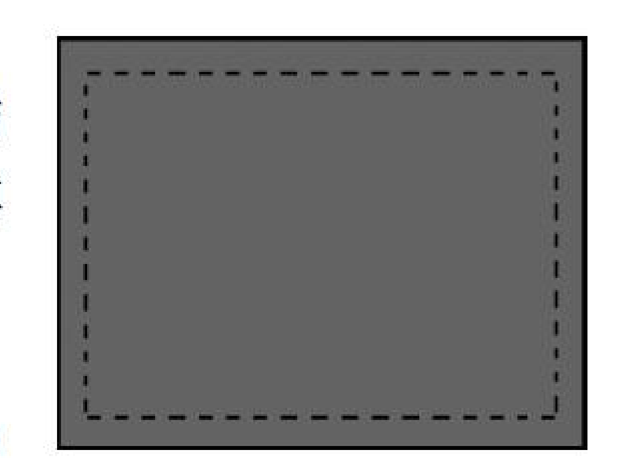


图 10-12　膨胀运算结果

膨胀运算的另一种等价定义如下：

(1)　$$X \oplus S = \bigcap \{X + s \mid s \in S\} \qquad (10\text{-}4)$$

(2)　$$X \oplus S = \bigcap \{S + x \mid x \in S\} \qquad (10\text{-}5)$$

上述各式中，式(10-3)和式(10-4)在进行算法设计时更方便，而式(10-5)突出了算法的几何特性。根据定义(10-3)可知，腐蚀和膨胀运算对集合运算的分配律只有在特定情况下才成立，在应用时应注意。另外，用腐蚀和膨胀运算也可以实现图像的平移。若自定义结构元素时选择不在原点的某一点作为结构元素，则得到的图像形状没有任何改变，只是位置发生了移动。

根据腐蚀和膨胀的定义，腐蚀和膨胀运算具有对偶性，其对偶性可用如下表达式表示：

$$(A \oplus S)^C = A^C \Theta S^V$$
$$(A\Theta S)^C \Theta = A^C \oplus S^V$$

例 10-4：腐蚀与膨胀对偶运算示例

二值腐蚀和膨胀运算可以用如图 10-13 所示进行解释。图 10-13(a)和图 10-13(b)为 A 和 S，图 10-13(c)为 A^C，图 10-13(d)为 S^V，$A \oplus S$ 的运算结果如图 10-13(e)所示，而 $A \Theta S$ 的运算结果如图 10-13(f)所示，$A^C \Theta S^V$ 的运算结果如图 10-13(g)所示，$A^C \oplus S^V$ 的运算结果如图 10-13(h)所示，图解中，深色点在膨胀结果中表示膨胀出来的点，而在腐蚀结果中表示腐蚀掉的点。

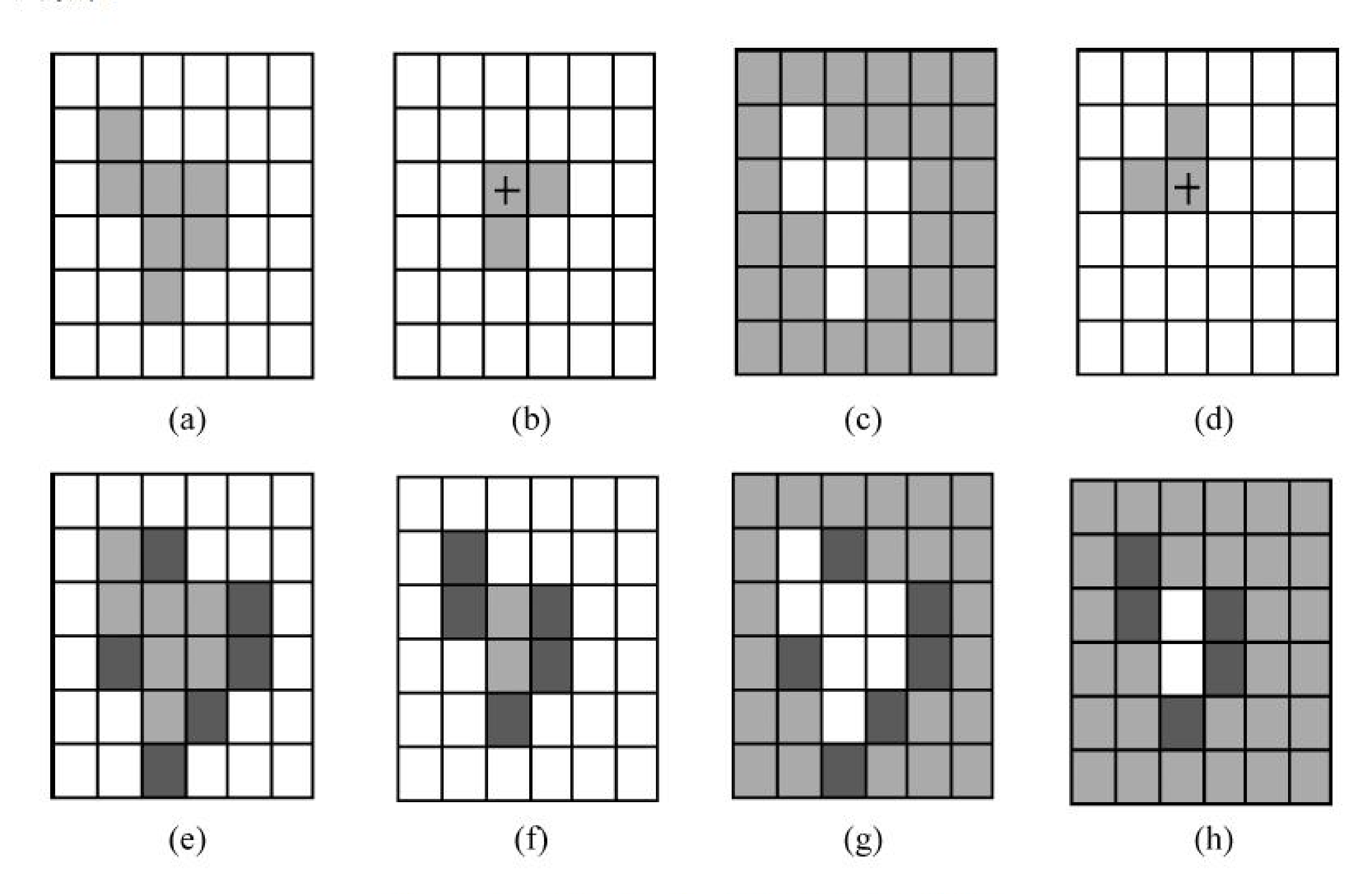


图 10-13　腐蚀与膨胀运算对偶性图解

10.2.3　开运算与闭运算

1. 基本概念

若结构元素为一个小的圆形，则膨胀可填充图像中比结构元素小的孔洞以及图像边缘处的小凹陷，使图像扩大；而腐蚀运算可以消除图像边缘的某些小区域，并将图像缩小，其补集扩大。但是，膨胀和腐蚀并非互逆运算，因此它们可以进行级联使用。在腐蚀和膨胀运算的基础上，可以构造数学形态学的其他运算算子，它们由腐蚀与膨胀两个基本运算的复合运算与集合操作(并、交、补等)组合成的所有运算构成。如可先对图像进行腐蚀再进行膨胀，或先对图像进行膨胀然后再进行腐蚀(同一个结构元素)。前者称为开运算(又称为开启运算)，后者称为闭运算(又称为闭合运算)。开运算和闭运算是形态学中两个非常重要的组合运算。

对图像 X 及结构元素 S，用符号 $X \circ S$ 表示 S 对图像 X 作开运算，用符号 $X \cdot S$ 表示 S 对图像 X 作闭运算，它们的定义为

$$X \circ S = \{(X \Theta S) \oplus S\} \tag{10-6}$$

$$X \cdot S = \{(X \oplus S) \Theta S\} \tag{10-7}$$

由式(10-6)和式(10-7)可知，$X \circ S$ 是指先进行腐蚀运算后进行膨胀运算的复合运算，可视为对腐蚀图像 $X \Theta S$ 再进行膨胀运算进行恢复，而 $X \cdot S$ 是指先进行膨胀运算后进行腐蚀运算的复合运算，可看作是对膨胀图像 $X \oplus S$ 再用腐蚀运算进行恢复。需要指出的是，这

一恢复不是信息无损的，即它们通常不等于原始图像 X。由开运算的定义式，可以得出

$$X \circ S = \bigcap \{S+x \mid S+x \subseteq X\} \tag{10-8}$$

因而 $X \circ S$ 是由所有 X 中与结构元素 S 完全相等的子集的并组成的。或者说，对 $X \circ S$ 中的每一个点 x，均可找到某个包含在 X 中的结构元素 S 经过平移(平移结果为 $S+y$)，使得

$$x \in S+y$$

即 x 在集合 X 的附近具有不小于 S 的几何结构，而对于 X 中不能被 $X \circ S$ 恢复的点，其附近的几何结构比 S 都要小。上述分析说明，$X \circ S$ 是一个基于几何结构的滤波器。

例 10-5：开运算实例

本示例是一道典型的开运算的运算原理图解题，如图 10-14 所示。图 10-14(a)为原始图像 X，图 10-14(b)为结构元素 S，图 10-14(c)为开运算结果。从该例可以观察到，集合 X 经过开运算后，图像内细小的连接被消除了。

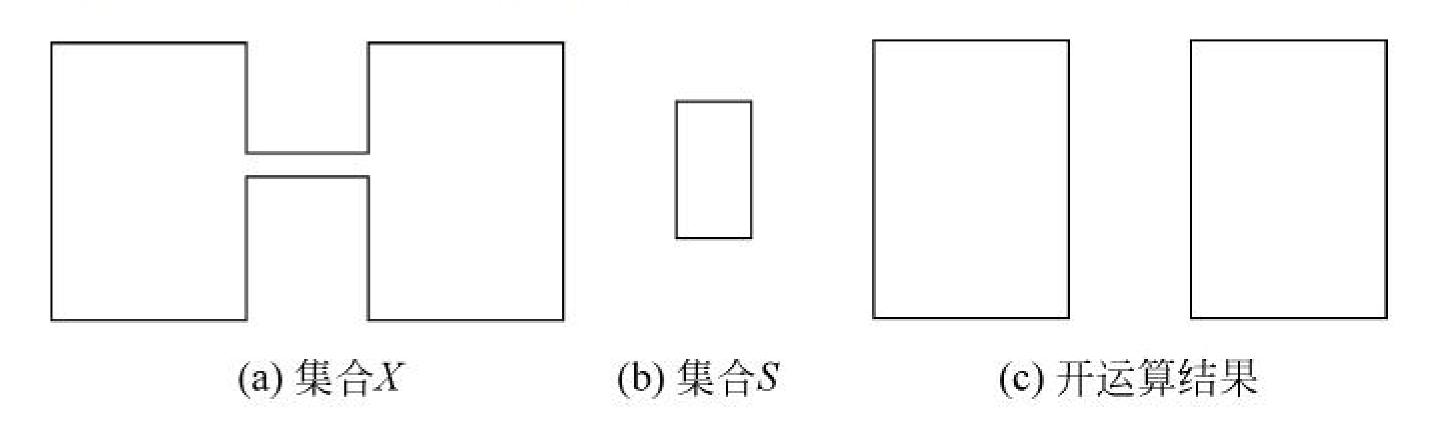


图 10-14　开运算实例图解

例 10-6：闭运算实例

本示例是一道典型的闭运算原理图解题，如图 10-15 所示。图 10-15(a)为原始圆形图像 X，图 10-15(b)为结构元素 S，图 10-15(c)为闭运算结果。该例表明，集合 X 经过闭运算后，图像内的小凹陷消失了。

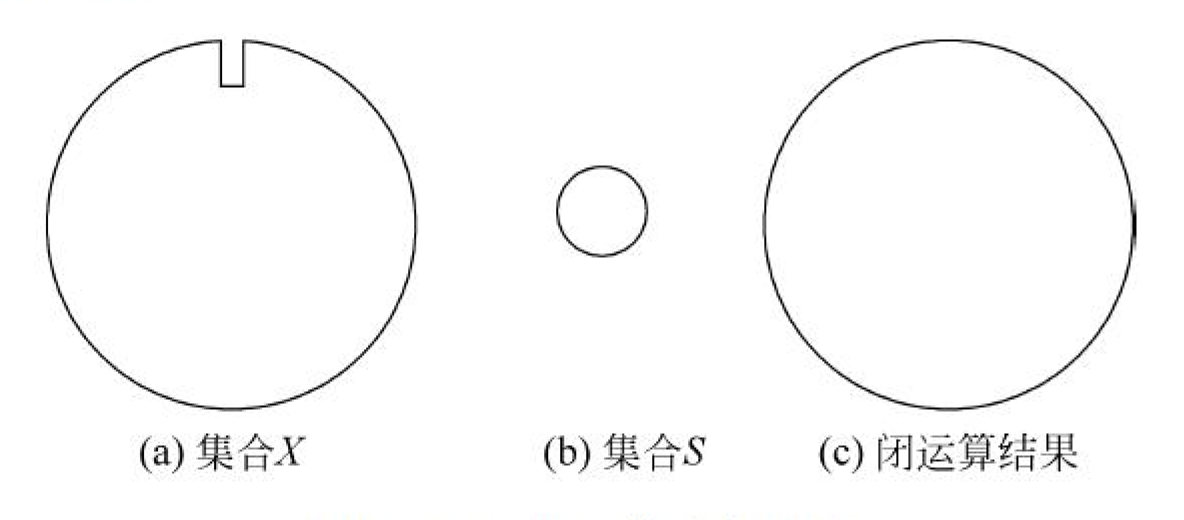


图 10-15　闭运算实例图解

通过上述开、闭运算的例题可以发现，开运算对图像的边界起到了平滑作用，因此，图 10-14(a)的图像 X 被分开为两个矩形；而闭运算可实现对图像的凹陷部分进行适当的填补。例 10-5 和例 10-6 表明，采用不同的结构元素，可提取到图像的不同特征。开、闭运算不受原点是否在结构元素之中的影响。使用不同的结构元素，开、闭运算的特性将得到进一步认识。

由腐蚀和膨胀的对偶性，可得出开、闭运算也是一对对偶变换，即

$$(X^C \circ S)^C = X \bullet S \tag{10-9}$$

$$(X^C \bullet S)^C = X \circ S \tag{10-10}$$

因此,闭运算的几何意义可以由补集的开运算的几何意义得出。

2. 开闭运算的代数性质

由于开、闭运算是定义在腐蚀和膨胀运算的基础上,因此根据腐蚀和膨胀运算的代数性质,可以得出如下性质:

1）对偶性

$$(X^C \circ S)^C = X \bullet S; \quad (X^C \bullet S)^C = X \circ S \tag{10-11}$$

2）扩展性

$$X \circ S \subseteq X \subseteq X \bullet S \tag{10-12}$$

即开运算使原图像缩小,而闭运算使原图像扩大。扩展性又称为收缩性。

3）单调性

若 $X \subseteq Y$,则

$$X \bullet S \subseteq Y \bullet S; \ X \bullet S \subseteq Y \circ S \tag{10-13}$$

若 $Y \subseteq Z, Z \bullet Y = Z$,则

$$X \bullet Y \subseteq X \bullet Z \tag{10-14}$$

根据单调性可知,若扩大后的结构元素对原结构元素开运算不变,则结构元素的扩大可保持单调特性。

4）平移不变性

$$\begin{aligned}(X+h) \bullet S &= (X \bullet S)+h, (X+h) \circ S = (X \circ S)+h \\ X \bullet (S+h) &= X \bullet S, X \circ (S+h) = X \circ S\end{aligned} \tag{10-15}$$

5）等幂性

$$(X \bullet S \bullet)S = X \bullet S, \quad (X \circ S) \circ S = X \circ S \tag{10-16}$$

开、闭运算的等幂性表明:经过一次滤波就能将所有特定结构元素的噪声滤除干净,但重复运算不再具有滤波效果。这一性质与中值滤波、线性卷积等性质的滤波特性不同。

6）开、闭运算与集合的关系

(1) $\left(\bigcup_{i=1}^{n} X_i\right) \circ S \supseteq \bigcup_{i=1}^{n} (X_i \circ S)$,即并集的开运算包含开运算的并集;

(2) $\left(\bigcap_{i=1}^{n} X_i\right) \circ S \subseteq \bigcap_{i=1}^{n} (X_i \circ S)$,即交集的开运算包含于开运算的交集中;

(3) $\left(\bigcup_{i=1}^{n} X_i\right) \bullet S \supseteq \bigcup_{i=1}^{n} (X_i \bullet S)$,即并集的闭运算包含闭运算的并集;

(4) $\left(\bigcap_{i=1}^{n} X_i\right) \bullet S \subseteq \bigcap_{i=1}^{n} (X_i \bullet S)$,即交集的闭运算包含于闭运算的交集中。

在对多个图像进行开、闭运算时,运用上述集合方面的性质可以显著提高开、闭运算的效率。

10.2.4 击中与击不中

数学形态学中,除膨胀、腐蚀、开运算和闭运算之外,击中/击不中变换(HMT)也是形态

学的一种基本运算，HMT变换可以同时探测图像的内部和外部。在研究图像中的目标物体与图像背景之间的关系时，HMT能够得到很好的效果，所以击中/击不中变换常用于解决目标图像识别和模式识别等领域的问题。

结构元素是形态学变换中的基本元素，是为了探测图像的某种结构信息而设计的特定形状和尺寸的图像，是一种收集图像结构信息的探针。结构元素除圆形、方形、线型等类型外，还有基于二值的、灰度值的、模糊集的等多种结构元素。在击中与击不中变换中将结构元素分解为两部分，一个定义为前景结构元素，一个定义为背景结构元素。具体定义如下：

设 A 为待处理图像，B 为结构元素，且由不相交的两部分 B_1 和 B_2 组成，且满足如下条件：

$$B = B_1 \cup B_2, \quad 且\ B_1 \cap B_2 = \phi$$

则 A 被 B 击中的定义为

$$A \odot B = \{x \mid B_1 + x \subseteq X \wedge B_2 + x \not\subset X\} \tag{10-17}$$

上式中的符号“∧”表示两个条件应同时满足。显然，当 $B_2 = \varnothing$ 时，击中变换就演变成了腐蚀运算。

式(10-17)也可以定义为

$$A \odot B = (A \Theta B_1) \cap (A^C B_2) \tag{10-18}$$

腐蚀运算的结果是结构元素 S 平移 x 但仍包含在输入图像 A 内部的所有结构元素的原点的集合，对于HMT变换，当且仅当结构元素 B_1 平移到某一点可以填入 A 的内部，且 B_2 平移到该点时可以填入 A 的外部时，该点才能在HMT变换的结果中输出。由于需要精确匹配，因此该算法对识别的要求很高，若图像包含未知噪声则会出现较大的误差，处理效果将受到影响。

10.3 灰度形态学

二值形态学的一些基本理论可以推广到灰度形态学。如腐蚀、膨胀、开和闭等基本运算都可方便地推广到灰度图像空间，只是灰度形态学的运算对象不再被视为集合而看作图像函数。

10.3.1 灰度腐蚀

设 $f(x,y)$ 是输入图像，$b(x,y)$ 是结构元素。用结构元素 $b(x,y)$ 对输入图像 $f(x,y)$ 进行灰度腐蚀简记为 $f\Theta b$，其定义为

$$(f\Theta b)(s,t) = \min\{[f(s+x,t+y) - b(x,y) \mid (s+x),(t+y) \in D_f,(x,y) \in D_b\} \tag{10-19}$$

式中，D_f 和 D_b 分别是 $f(x,y)$ 和 $b(x,y)$ 的定义域。

这里要求 $(s+x,t+y)$ 和 (x,y) 在其定义域内，之所以这样要求，与二值腐蚀运算的定义中要求结构元素完全包括在被腐蚀集合中类似。

若结构元素是一维函数，则式(10-19)可简化为

$$(f\Theta b)(s) = \min\{[f(s+x) - b(x) \mid (s+x) \in D_f, x \in D_b\} \tag{10-20}$$

例 10-7：灰度腐蚀运算示例

如图 10-16(a)和(b)分别为图像 f 和结构元素 b，用 b 对 f 进行腐蚀操作的结果如图 10-16(c)所示。

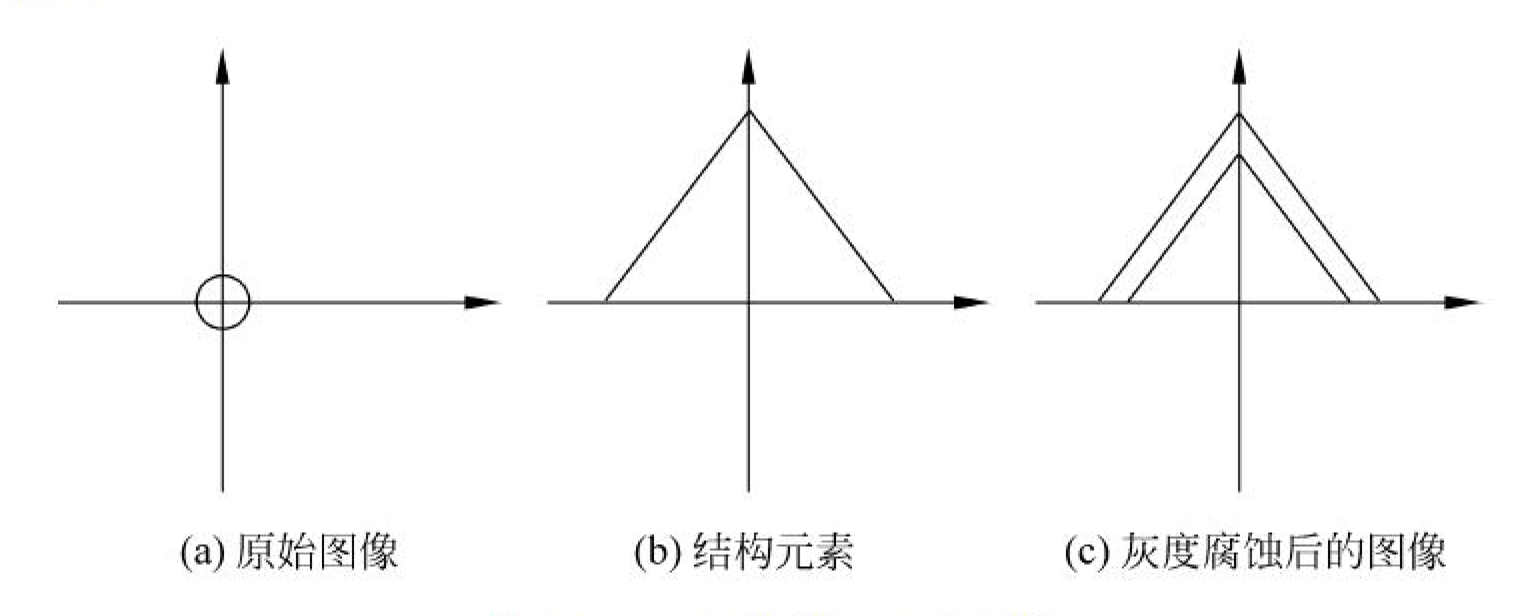


图 10-16　灰度腐蚀后的图像

腐蚀运算是在由结构元素确定的邻域中选取 $f\Theta b$ 的最小值，因此，对灰度图像的腐蚀操作有两类效果：

(1) 若结构元素的值都为正，则处理后的图像会比输入图像暗。

(2) 若输入图像中亮细节的尺寸比结构元素小，则影响将被减弱，减弱的程度取决于这些亮细节周围的灰度值和结构元素的形状与幅值。

10.3.2　灰度膨胀

用结构元素 $b(x,y)$ 对输入图像 $f(x,y)$ 进行灰度膨胀记为 $f\oplus b$，其定义为

$$(f\oplus b)(s,t)=\max\{[f(s+x,t+y)+b(x,y)\mid(s+x),(t+y)\in D_f,(x,y)\in D_b\}\tag{10-21}$$

式中，D_f 和 D_b 分别是 $f(x,y)$ 和 $b(x,y)$ 的定义域。

这里要求 $(s+x,t+y)$ 和 (x,y) 在其定义域内，类似于在二值膨胀定义中要求两个运算集合至少有一个非零元素相交。式(10-21)与二维离散函数的卷积的形式很类似，仅仅是式(10-21)用 max 运算代替了卷积的求和，用加法代替了卷积的相乘。若结构元素为一维函数，则式(10-21)可简化为

$$(f\oplus b)(s)=\max\{[f(s+x)+b(x)\mid(s+x)\in D_f,x\in D_b\}\tag{10-22}$$

和卷积相似，若 b 为正，$f(x,y)$ 移向右边；若 b 为负，$f(x,y)$ 移向左边。要求 $(s+x)$ 和 x 在其定义域内，是为了使 f 和 b 的交集非空。

对灰度图像的膨胀运算有两类效果：

(1) 若结构元素的值都为正，则处理后的图像会比输入图像亮。

(2) 根据输入图像中暗区细节的灰度值及其形状相对于结构元素的关系，原图像中的暗区在膨胀中或被消减或被完全消除。

10.3.3　灰度开运算与闭运算

数学形态学中，灰度开、闭运算的定义与二值数学形态学中的开、闭运算是一致的。因此，开、闭运算的定义可以类似地给出，即对于开运算先腐蚀后膨胀，对闭运算先膨胀后腐蚀。

1. 开运算

用结构元素 $b(x,y)$ 对灰度图像 $f(x,y)$ 进行开运算记为 $f\circ b$，其定义为

$$f \circ b = (f \Theta b) \oplus b \tag{10-23}$$

灰度开运算处理的结果是小的、明亮的细节尺寸变小了，而暗区的效果没有明显的变化。

2. 闭运算

用结构元素 $b(x,y)$ 对灰度图像 $f(x,y)$ 进行闭运算记为 $f\bullet b$，其定义为

$$f \bullet b = (f \oplus b) \Theta b \tag{10-24}$$

灰度闭运算处理的结果是小而暗的细节尺寸变小了，而明亮区域的效果没有明显的变化。

10.4 数学形态学的应用

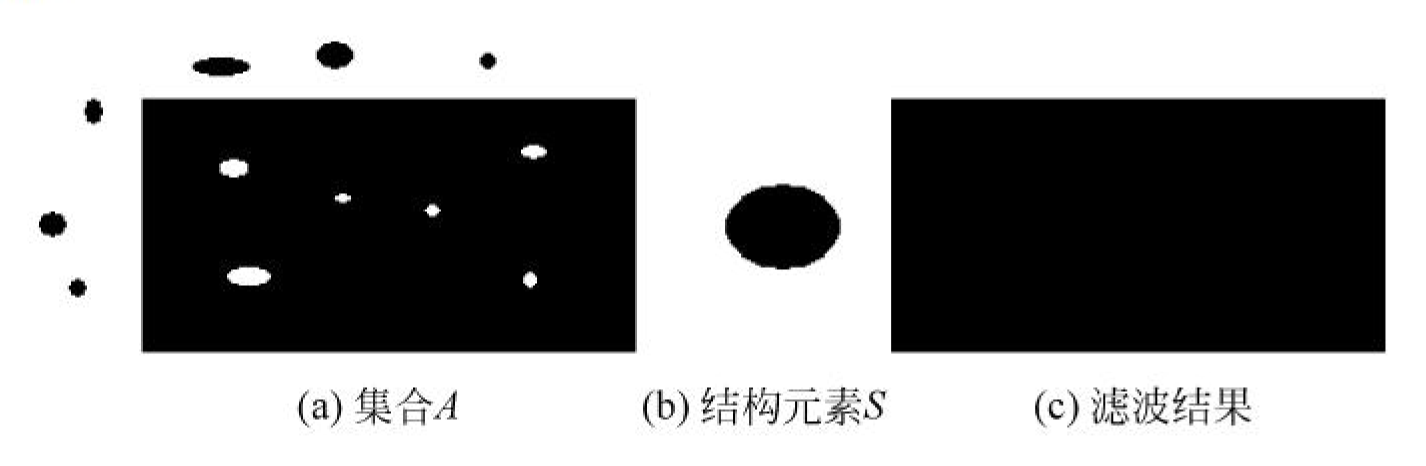
前面介绍了二值形态学和灰度形态学的腐蚀、膨胀、开、闭等基本运算及相关性质，通过这些运算的组合可以得到很多二值形态学和灰度形态学的实用算法。灰度形态学的主要算法有灰度形态学平滑、纹理分割、形态学梯度、高帽(top-hat)及低帽(bot-hat)变换等。数学形态学运算具有广泛的应用，在应用形态学方法处理实际图像时，通常需要对输入图像进行预处理，以便适合于算法要求。

10.4.1 二值形态学平滑滤波

根据开、闭运算的定义，对于二值图像先进行开运算，再进行闭运算可以实现图像的平滑滤波，消除图像的部分噪声(如人为的亮点和暗点等)。

例 10-8：二值形态学滤波示例

如图 10-17(a)集合(图像)A 存在若干人为噪声，包括亮点和暗点。例如，采用图 10-17(b)所示圆形结构元素，对集合 A 先进行开运算，然后再进行闭运算，可消除噪声。处理结果如图 10-17(c)所示。



(a) 集合A　　(b) 结构元素S　　(c) 滤波结果

图 10-17　二值形态学滤波示意图

图 10-17 给出了利用圆形结构元素进行二值滤波的结果，实际上形态学滤波处理也可以应用于灰度图像。利用适当的结构元素，灰度开运算可用于滤除图像的某些噪声。

10.4.2 图像细化

采用图像细化方法得到图像内的细化结构是图像处理中的常用方法，将图像细化成骨架是一种重要的结构形状表示法。例如，在进行字符识别前，往往要先对字符进行细

化处理，求出字符的细化结构。因此，分析二值图像的细化结构是图像处理的一个基本问题，在图像识别或数据压缩时，常常用到分析图像的细化结构，骨架是目标图像的重要拓扑描述，具有广泛的应用。骨架既可以采用几何方法也可以采用数学形态学方法求出。

1. 几何方法

图像骨架形状的描述方法由 Blum 最先提出，如图 10-18 所示的矩形图像，矩形中的虚线为图像的骨架。该骨架由一条水平线和四根斜线组成，其中，水平骨架由区域内的同心圆的圆心轨迹构成。四根斜线骨架也可采用类似的方法求出，以矩形右上方所示的斜线骨架(右上方逆时钟 45°虚线)为例，该骨架线由一系列大小不同且与矩形右上角相邻两矩形边的内切圆的圆心轨迹组成。

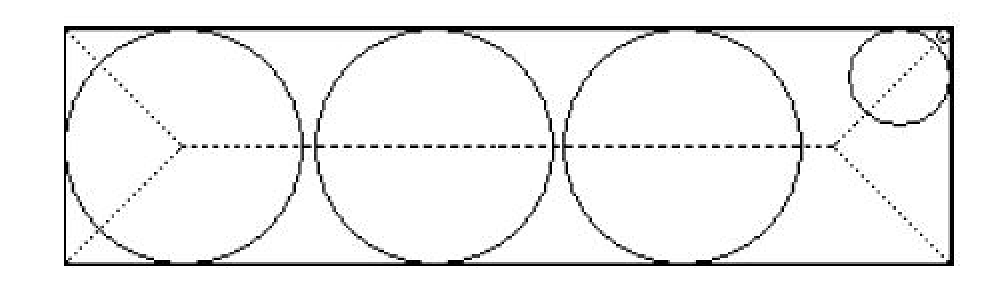


图 10-18　图像骨架的示意图

2. 形态学方法

除几何学方法之外，骨架也可以采用数学形态学方法进行定义和求解。设图像 A 的骨架用 $S(A)$表示，则图像骨架可以用腐蚀和开运算表示为

$$S(A)=\bigcup_{k=0}^{N} S_k(A) \tag{10-25}$$

$$S_k(A)=(A\Theta kB)-(A\Theta kB)\circ B \tag{10-26}$$

式中，$S_k(A)$表示图像 A 的骨架子集；$(A\Theta kB)$表示采用结构元素 B 对 A 进行连续 k 次腐蚀运算；N 是指 A 被腐蚀为骨架之前的最后一次腐蚀迭代运算之次数，因此 $N=\max\{k|A\Theta kB)\neq\varnothing$。

例 10-9：图像骨架提取运算示例

如图 10-19(a)为包含汉字“图像处理”的原始图像，对该图像进行骨架提取运算，结果如图 10-19(b)所示。

图像处理　图像处理

(a) 原始图像　(b) 骨架图像

图 10-19　骨架提取算法

图像 A 也可以根据 $S_k(A)$进行重构，重构公式为

$$A=\bigcup_{k=0}^{N}\{S_k(A)\oplus kB\} \tag{10-27}$$

上式中的 $S_k(A)\oplus kB$ 表示采用结构元素 B 连续 N 次对 $S_k(A)$进行膨胀运算。

10.4.3　图像粗化

图像粗化是相对于图像细化而言的，图像粗化可以用击中与击不中表示，图像粗化的公

式为

$$S(A) = A \cup (A \odot B) \tag{10-28}$$

根据上式可以看出,图像粗化是原始图像与击中原始图像的并集。

习题

1. 数学形态学具有哪些用途?

2. 若采用一个半径为 0.5cm 的圆形作为结构元素,对半径为 2cm 的圆进行腐蚀和膨胀运算,分析其结果。

3. 根据二值腐蚀运算的原理,给出编程实现腐蚀运算的步骤。

4. 根据二值膨胀运算的原理,给出编程实现膨胀运算的步骤。

5. 集合 A 和结构元素 S 的形状如图 10-20 所示,求用 S 对 A 进行腐蚀运算的结果。

6. 数学形态学的基本运算腐蚀、膨胀、开和闭运算各有何性质? 试比较其异同。

7. 集合 X 和结构元素 S 的形状如图 10-21 所示,求用 S 对 X 进行膨胀运算的结果。

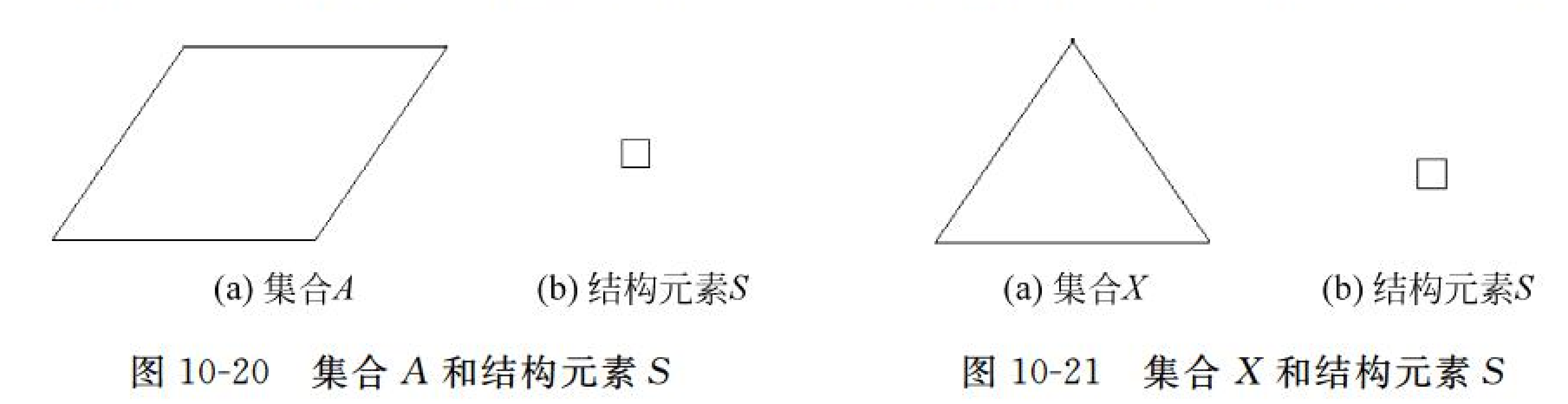


图 10-20 集合 A 和结构元素 S　　图 10-21 集合 X 和结构元素 S

8. 试编写一个程序,实现二值图像的腐蚀、膨胀及开、闭运算。

9. 试编写一个程序,实现灰度图像的腐蚀、膨胀运算。

10. 什么是图像的骨架? 试简述骨架提取的基本原理。

第11章 图像分割

CHAPTER 11

数字图像处理按其技术特征可以分为三个层次,即图像处理、图像分析和图像理解与识别。无论图像处理、分析还是图像理解与识别,一般都建立在图像分割的基础上,将图像中有意义的特征或应用所需要的特征信息提取出来。图像分割的最终结果是将图像分解成一些具有某种特征的单元,称为图像的基元。相对于整幅图像来说,这种图像基元更容易被快速处理。图像特征是指图像中可用作标志的属性,它可以分为统计特征和视觉特征两类。图像的统计特征是指一些人为定义的特征,通过变换才能得到,如图像的直方图、矩、频谱等;图像的视觉特征是指人的视觉可直接感受到的自然特征,如区域的亮度、纹理和轮廓等。

图像分割是进行图像分析的关键步骤,也是进一步理解图像的基础。关于图像分割,每年都会发表很多研究文章,提出各种新的图像分割方法。不同种类的图像,不同的应用要求所需要提取的特征是不相同的,其特征提取方法也就不同,因此并不存在一种所谓普遍适用的最优方法。图像的边缘对人类的视觉系统具有重要的意义,它是人类判别物体的重要依据,是图像的最基本特征。所谓边缘(又称边沿),是指其周围像素灰度有阶跃变化或屋顶变化的那些像素的集合。边缘广泛存在于物体与背景之间、物体与物体之间、基元与基元之间。因此,它是图像分割所依赖的重要特征。

本章围绕图像分割的基本原理,介绍了常用的三种图像分割方法,包括阈值分割法、区域分割法和基于边缘的分割方法,同时还讨论了六种常用的边缘检测算子、轮廓提取方法和图像匹配法等图像分析与处理的重要基础算法。

11.1 图像分割的基本概念

顾名思义,图像分割的目的是将图像划分为若干具有一定意义的或一定目的的子区域。例如,一幅航空照片,可以分割为工业区、居民区、森林和湖泊等区域。可以逐个像素点为基础研究图像分割,也可以利用在规定邻域中的某些图像信息进行分割。分割的依据一般建立在相似性和非连续性两个基本概念的基础上。

11.1.1 图像分割的基本原理

图像分割是根据图像的组成结构和应用需求将图像划分成若干个互不相交的子区域的过程,这些子区域是某种意义下具有共同属性的像素的连通集合。如不同目标物体所占的图像区域、前景所占的图像区域等。连通是指集合中任意两个点之间都存在着完全属于该

集合的连通路径。

对于数字图像而言，如图 11-1 所示，连通包括 4 连通和 8 连通两种情况。4 连通指的是从区域内一点出发，可通过 4 个方向，即上、下、左、右移动的组合，在不超出区域的前提下，到达区域内的任意像素点；8 连通方法指的是从区域内一点出发，可通过左、右、上、下、左上、右上、左下、右下这 8 个方向的移动组合来到达区域内的任意像素点。

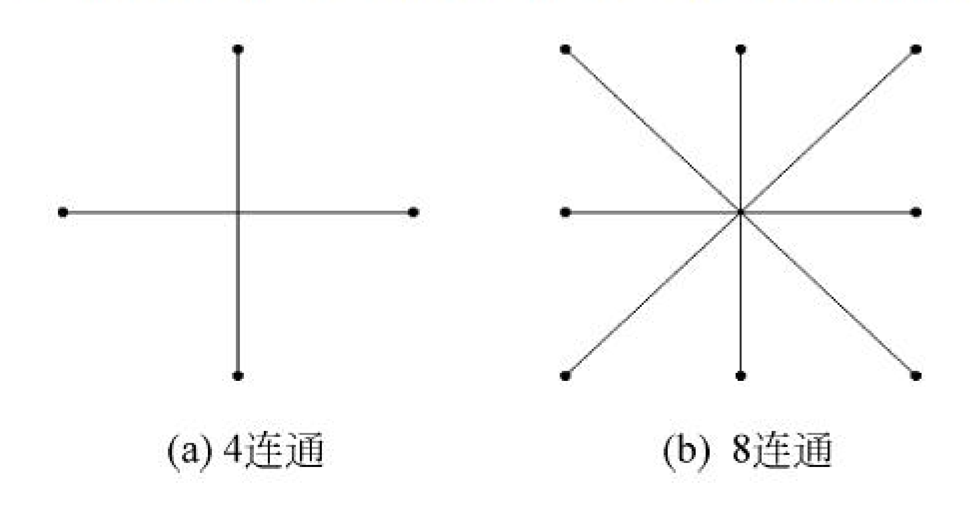


图 11-1　4 连通和 8 连通示意图

根据上述基本概念，可以给出图像分割的一般性定义，即图像分割是指将一幅离散数字图像信号 $f(m,n)$ 分割为若干相连的、非空子区域 $f_1, f_2, f_3, \cdots, f_n$，且满足如下均一性准则：

(1) $f_1 \cup f_2 \cup f_3 \cdots \cup f_n = f$。

(2) $\forall i$，当 $i=1,2,3,\cdots,n$ 时，f_i 是连通区域。

(3) $\forall i,j$，若 $i \neq j$ 时，$f_i \cap f_j = \varnothing$。

(4) $\forall f_i, P(f_i) = \text{True}$。

(5) $\forall i,j$，若 $i \neq j$ 时，$P(f_i \cap f_j) = \text{False}$。

该定义的条件(1)确保一幅图像经过分割以后，其全部子区域的总和应包括被分割图像的所有像素，即图像中的每一个像素都能分割到一个确定的子区域中。条件(2)保证了分割的各自区域内的像素是连通的，即分割得到的各子区域应该是一个连通的小区域。条件(3)确保了被分割图像的各个子区域是互不重叠的区域，或者说，被分割图像的任何一个像素不能同时属于两个或两个以上区域之中。

11.1.2　图像分割方法

图像分割就是把图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标的技术和过程，是由图像处理到图像分析的关键步骤。目前已经提出的图像分割方法很多，综合各种方法的实质，现有的图像分割方法主要分以下几种不同的途径：

(1) 基于阈值的分割方法，是一种最常用的并行区域技术，它是图像分割中应用数量最多的一类。阈值分割方法实际上是输入图像到输出图像基于某一阈值的门限值变换。

(2) 基于区域的分割方法，以相似性原则作为分割的依据，即根据图像的灰度、色彩、变换关系或组织结构等方面的特征相似来划分图像的子区域，并将各像素划归到相应物体或区域的像素聚类方法，即区域法。

(3) 基于边缘的分割方法，边缘检测是图像分割的一种重要途径，即检测灰度级或者结构具有突变的地方，表明一个区域的终结，也是另一个区域开始的地方。图像内不同的目标灰度不同，边界处一般有明显的边缘，利用此特征可以分割图像。

此外，还有一些基于特定理论的分割方法，从数学角度来看，图像分割是将数字图像划

分成互不相交的区域的过程，图像分割的过程也是一个标记过程，即把属于同一区域的像素赋予相同的编号。

这些方法是互补的，在有些场合适宜应用这一种分割方法，而另一些场合则适宜采用另一种分割方法，有时还要将它们有机地结合起来，以求得到更好的分割效果。

对图像分割算法的研究已有几十年的历史，目前基于各种理论已提出了上千种分割算法。尽管人们在图像分割方面做了许多研究工作，但由于尚无通用分割理论，因此，这些分割算法大都是针对具体问题的，并没有一种适合于所有图像的通用的分割算法。但可以指出的是，图像分割方法正朝着更快速、更精确的方向发展，通过各种新理论和新技术结合将不断取得突破和进展。

虽然图像分割没有唯一的、标准的、通用的分割方法，也没有一成不变、适应一切情况的所谓最优分割准则，必须根据具体图像和不同应用目的而采用合适的分割方法。但在已有的图像分割技术中，最常用的是基于阈值处理、基于区域和基于边缘的图像分割方法。

11.2 阈值分割法

阈值化分割是一种常用的、传统的图像分割方法，实现简单、计算量小、性能较稳定，目前在图像分割中已成为一种基本方法，得到了广泛的应用。该分割方法特别适用于目标和背景占据不同灰度级范围的图像，它不仅可以极大地压缩数据量，而且也大大简化了分析和处理步骤，因此在很多情况下，是进行图像分析、特征提取与模式识别之前的必要的图像预处理过程。

常用的图像分割方法是把图像灰度分成不同的等级，然后用设置灰度门限值(阈值)的方法确定有意义的区域或分割物体的边界。常用的阈值化处理就是图像的二值化处理，即选择一阈值，将图像转换为黑白二值图像，用于图像分割及边缘提取等处理之中。

图像阈值化处理的变换函数形式为

$$g(x,y)=\begin{cases}0 & f(x,y)\leqslant T\\ 255 & f(x,y)>T\end{cases} \tag{11-1}$$

显然，图像阈值化处理是一种阶梯函数，属于图像灰度级的非线性运算，该变换函数曲线如图 11-2 所示。它的功能是由用户指定一个阈值，如果图像中某个像素的灰度值大于该阈值，则将该像素的灰度值置为 255，否则将其灰度值置为 0。

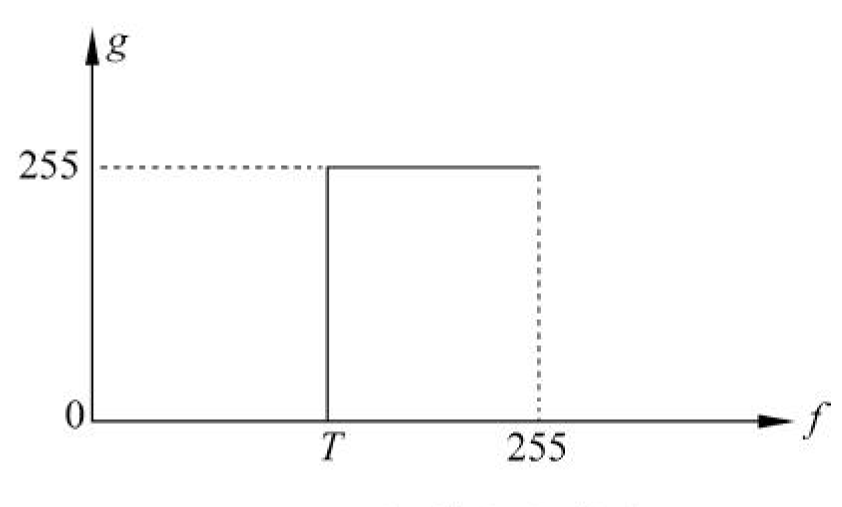


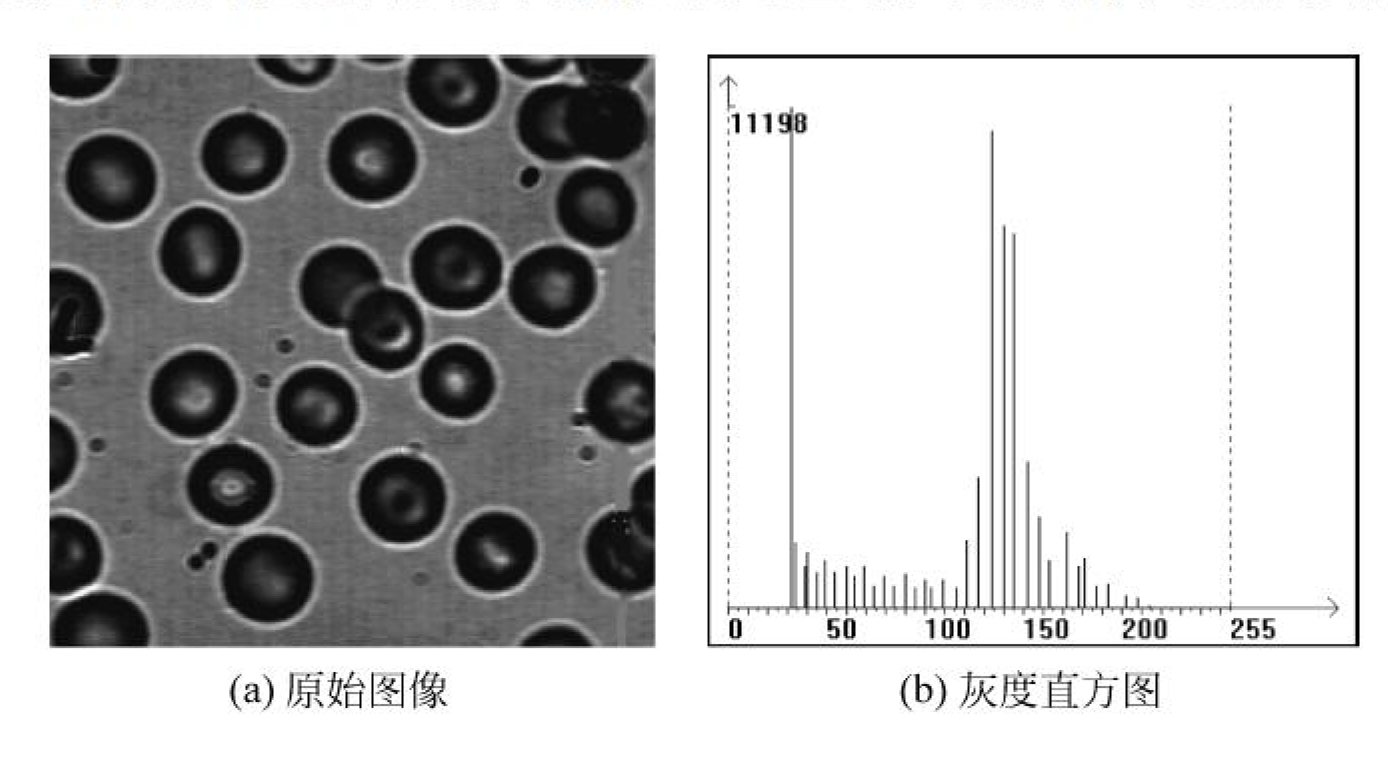
图 11-2　阈值变换曲线

阈值分割的优点是计算简单、运算效率较高、速度快。对运算速度和效率要求高的应用场合(如用于硬件实现)，它得到了广泛应用。目前，图像的阈值分割在很多领域得到了广泛应用。例如，在红外技术应用中，红外无损检测中红外热图像的分割，红外成像跟踪系统中目标的分割；在遥感图像处理中，常用于合成孔径雷达图像对目标进行分割；在医学图像中，常用于血液细胞图像的分割、磁共振图像的分割；在农业工程中，水果品质无损检测过程中水果图像与背景的分割等。

采用图像的阈值化分割过程中，应合理选择阈值，例 11-1 表明，阈值选取的合理性对处理结果的影响很大。

例 11-1：阈值分割法实验

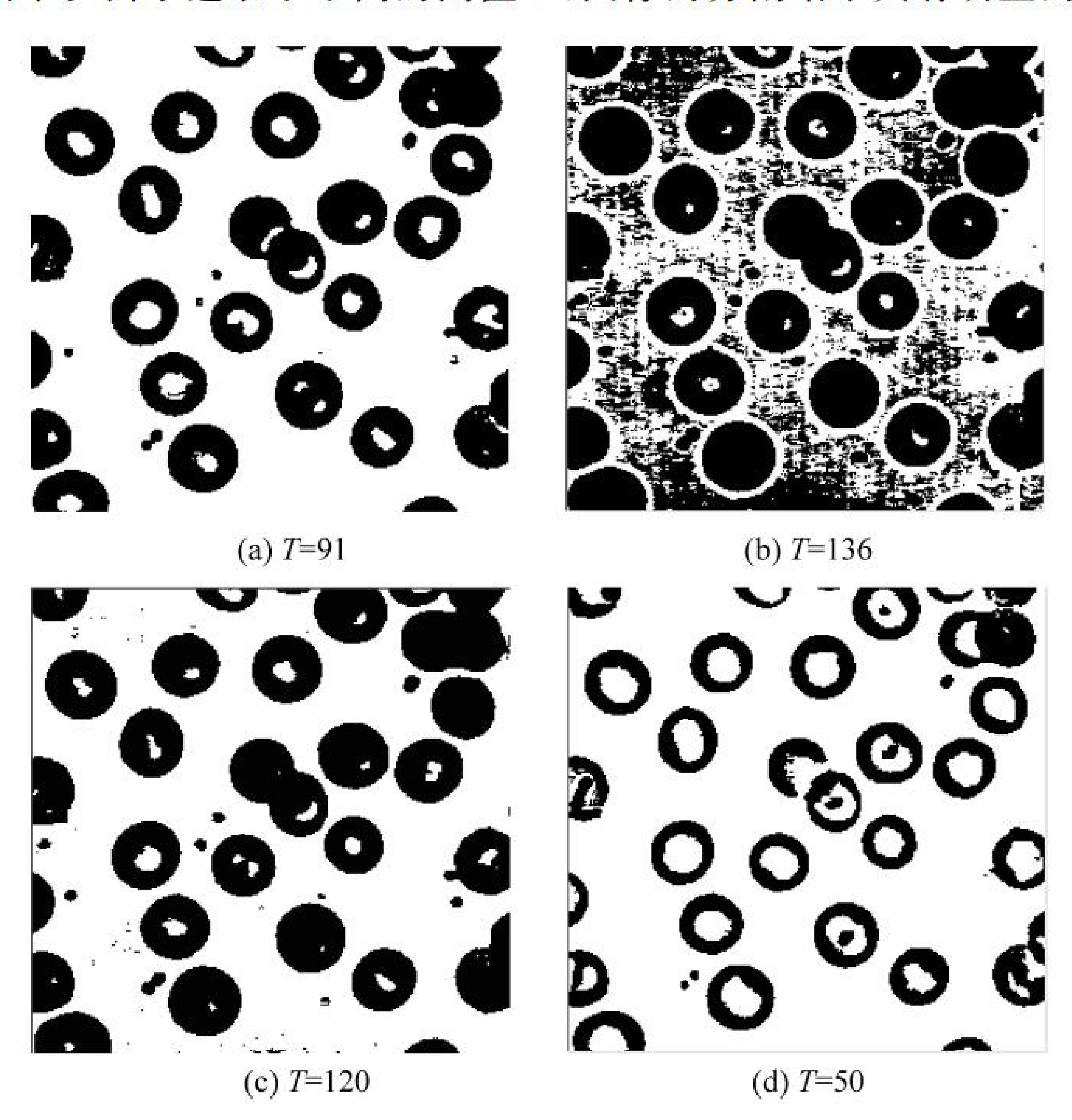
阈值分割法的关键是如何选择阈值。如图 11-3 所示，对血液细胞图像进行阈值分割，图 11-3(a)为原始图像，图像中的目标为细胞；图 11-3(b)为原始图像的灰度直方图。分析该直方图可知，该直方图具有双峰特性，图像中的目标分布在较暗的灰度级上形成一个波峰，图像背景分布在较亮的灰度级上形成另一个波峰。因此，从理论上讲，以直方图双峰之间的低谷处灰度值作为阈值进行图像的阈值化处理，便可将目标和背景分割开来。



(a) 原始图像　　(b) 灰度直方图

图 11-3　阈值分割法原理

根据原始图像的灰度直方图，如图 11-3(b)所示，在 50～136 之间选取不同阈值进行分割实验和分割效果对比。图 11-4(a)为选取阈值 $T=91$ 的分割结果，图 11-4(b)为选取阈值 $T=136$ 的分割结果，图 11-4(c)为选取阈值 $T=120$ 的分割结果，图 11-4(d)为选取阈值 $T=50$ 的分割结果。由于选取了不同的阈值 T，图像的分割结果具有明显的差别。


(a) T=91　　(b) T=136

(c) T=120　　(d) T=50

图 11-4　不同阈值对阈值化结果的影响

根据该实验结果，基于阈值的图像分割过程中，若阈值选取过大，则会提取图像的多余部分；若阈值过小，则又会丢失所需的部分。因此，阈值的选取对图像的分割具有非常重要的作用。

当图像灰度直方图的双峰特性不明显，即图像中的目标部分和背景之间亮度差较小时，直接用直方图就不太容易确定一个合适的阈值。此时，可用最佳阈值法、判别分析法或其他合理的方法来确定阈值。

11.2.1 最佳阈值法

设图像由物体和背景两部分组成，物体像素的灰度级具有正态概率密度函数 $p(z)$，其均值为 μ，方差为 σ^2。背景像素的灰度级也具有正态概率密度函数 $q(z)$，其均值为 ν，方差为 τ^2。物体占总图像面积的比例为 λ，背景所占面积比为 $1-\lambda$，因此，该图像总的灰度概率密度函数为

$$\lambda p(z)+(1-\lambda)\,q(z) \tag{11-2}$$

令图像阈值为 t，并且将小于 t 的全部点称为目标物体点，而将不小于 t 的全部点称为背景点。那么，设将背景点错归为物体点的概率为 $Q_1(t)$，将物体点错归为背景点的概率为 $Q_2(t)$，因而有

$$Q_1(t)=\int_{-\infty}^{t} q(z)\mathrm{d}z \tag{11-3}$$

$$Q_2(t)=\int_{t}^{\infty} p(z)\mathrm{d}z=1-p(t) \tag{11-4}$$

总的错分概率为

$$\lambda Q_2(t)+(1-\lambda)Q_1(t)=\lambda[1-p(t)]+(1-\lambda)Q_1(t) \tag{11-5}$$

显然，使上式最小的阈值 t 为最佳阈值。因此，对上式微分可得

$$(1-\lambda)q(t)=\lambda p(t) \tag{11-6}$$

由于 $p(t)$ 和 $q(t)$ 都服从正态分布，因而有

$$p(t)=\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(t-\mu)^2}{2\sigma^2}} \tag{11-7}$$

$$q(t)=\frac{1}{\sqrt{2\pi}\tau}\mathrm{e}^{-\frac{(t-\nu)^2}{2\tau^2}} \tag{11-8}$$

将以上两式代入式(11-6)并取自然对数可得

$$\ln\sigma+\ln(1-\lambda)-\frac{(t-\nu)^2}{2\tau^2}=\ln\tau+\ln\lambda-\frac{(t-\mu)^2}{2\sigma^2}$$

$$\tau^2\,(t-\mu)^2-\sigma\,(t-\nu)^2=2\sigma^2\tau^2\ln\frac{\tau\lambda}{\sigma(1-\lambda)}$$

根据上式，若 $\lambda=0.5$，$\tau=\sigma$，则可以得出

$$t=\frac{\mu+\nu}{2} \tag{11-9}$$

11.2.2 判别分析法

判别分析法确定最佳阈值的原则是进行阈值处理后，被分离的像素类之间的类间方差最大。判别分析法需要计算灰度直方图的 0 阶矩和 1 阶矩，它是图像阈值化处理中常用的

自动确定阈值的方法。

设数字图像的灰度级数为 L，总像素数为 N，灰度值为 i 的像素数以 N_i 表示，则灰度级为 k 的灰度分布的 0 阶矩及 1 阶矩的定义分别如下：

0 阶矩为

$$\omega(k) = \sum_{i=0}^{k} \frac{N_i}{N} \tag{11-10}$$

1 阶矩为

$$\omega(k) = \sum_{i=0}^{k} \frac{iN_i}{N} \tag{11-11}$$

当 $k=L-1$ 时，$\omega(L-1)=1$；$\mu(L-1)=\mu_A$，其中，μ_A 为图像的平均灰度值。

设图像的 $M-1$ 个阈值为 $0\leqslant k_1<k_2<k_3<\cdots<k_{M-1}\leqslant L-1$，将图像分割成 M 个灰度值的类 C_j

且 $C_j\in[k_{j-1}+1,\cdots,k_j]$　$k_0=0,k_M=L,j=1,2,3,\cdots,M$

则各类 C_j 的发生概率为

$$\omega_j = \omega(k_j) - \omega(k_{j-1})$$

而相应的概率平均值为

$$\mu_j = \frac{\mu(k_j) - \mu(k_{j-1})}{\omega(k_j) - \omega(k_{j-1})} \tag{11-12}$$

其中

$$\omega(0) = 0, \quad \mu(0) = 0$$

因此，可得各类的类间方差为

$$\sigma^2(k_1,k_2,k_3,\cdots,k_{M-1}) = \sum_{j=0}^{M-1}\omega_j\,(\mu_j - \mu_T)^2$$

将使上式的 σ^2 值为最大的阈值组$(k_1,k_2,k_3,\cdots,k_{M-1})$作为 M 值化的最佳阈值组。若分割成 2 类，即取 M 为 2，则可求出二值化的阈值。

对于复杂图像，在许多情况下对整幅图像用单一阈值不能给出理想的分割结果。例如，若为光亮背景上的暗物体图像，由于照射光的不均匀，虽然物体与背景始终有反差，但在图像的某一部分物体和背景都比另一部分亮。因此，在图像的一部分能把物体和背景精确地分开的阈值，对另一部分而言，可能会把太多的背景也作为物体进行分割。克服这一缺点有如下一些方法：如果已知图像上的位置函数用来描述不均匀照射，则可以先设法应用灰度级校正方法进行校正，然后采用单一阈值进行分割；另一种方法是将图像分成子区域，对每一子区域设置局部阈值。但是，如果某一子区域仅仅含有物体或仅有背景，那么该子区域就找不到阈值。这时，可以根据邻近子区域的局部阈值通过内插法求得该子区域的一个阈值。

在确定阈值时，如果阈值过高，偶然出现的物体像素点就会被认作背景；如果阈值过低，则会发生相反的情况。克服这种情况的方法是使用两个阈值。例如，$t_1<t_2$，将灰度值超过 t_2 的像素分类为核心物体点，而灰度值超过 t_1 的像素，仅当它们紧靠核心物体点时才认为是物体点。t_2 的选择要使每个物体有一些像素灰度级高于 t_2，而背景不含有这样的像素。同时，应选择 t_1 使每个物体像素点具有高于 t_1 灰度级。如果只使用 t_2，则物体总是分割得不完整；如果只使用 t_1，则会有许多背景像素被错分为物体像素。但是，如果同时使用 t_1 和 t_2 两个值就能把背景与物体很好地分割开来。当然，如果物体与背景的对比是鲜明的，就不

必使用这种方法。

此外,如果存在一个阈值 t_2,使得每个物体的像素灰度级高于 t_2,而背景不包含这种像素,可对图像设置阈值 t_2,然后检查高于阈值像素的区域,目的是寻找一个局部阈值,以便在每个类似区域中把物体和背景分开。如果这些物体相当小,并且不太靠近在一起时,这种方法比较适用。所使用的区域应足够大,以保证它们既包含物体像素,也包含背景像素,这样就可以使区域的直方图是双峰的。

有时需要寻找一幅图像的局部最大点,即提取比附近像素有较高的某种局部性质值的像素。一般来讲,也要求这些点具有高于一个低阈值 t_1 的值,一旦超过 t_1,不管它的绝对值大小如何,一切相对的最大值都被采纳。因此,可把寻找局部最大值看作为局部设置阈值的极端情况。在对图像进行匹配运算或检测界线时可采用这种方法。

11.3 区域分割法

基于区域的图像分割方法主要包括区域生长和区域分裂与合并两种典型的方法。区域生长法是从单个像素出发,逐渐合并周围像素,最后形成所需要的分割结果;分裂与合并法是从整幅图像出发,逐渐分裂或者合并,最后形成所需要的分割结果。

11.3.1 区域生长法

图像分割的目的是将一幅图像划分成一些子区域,因此,最直接的方法就是将一幅图像分成满足某种判据的子区域,也就是说,将点组成区域。为了实现分组,首先需要确定子区域的数目,其次要确定一个各子区域相互区别的特征,最后还要产生有意义分割的相似性判据。

将图像分割成子区域的一种方法叫区域生长或区域生成法,其基本思想是将图像中满足某种相似性准则的像素点集合起来构成区域。具体先对每个需要分割的区域找一个种子像素作为生长的起点,然后将种子像素周围邻域中与种子像素有相同或相似性质的像素(根据某种事先确定的生长或相似准则来判定)合并到种子像素所在的区域中。将这些新像素当作新的种子像素继续进行上面的过程,直到再没有满足条件的像素可被包括进来。这样一个区域就长成了。

假定子区域的数目以及在每个子区域中单个点的位置已知,则从一个像素点(种子点)开始,在各个方向上,如 4 邻域或 8 邻域向外扩展以生成子区域。相似性准则可以是灰度级、彩色、组织、梯度或其他特性,相似性测度可以根据所确定的阈值来判定。方法是从满足检测准则的点开始,在各个方向上生长子区域,当其邻近点满足检测准则就并入小块区域中。当新的点被合并后再用新的子区域重复这一过程,直到没有可接受的邻近点时生成过程便终止。

例 11-2:简单区域生长示例

本示例是区域生长法的一个算法实例,如图 11-5 所示,图像中的种子像素点和被合并的像素点均用下划线标出。其中图 11-5(a)是输入图像,像素点 9 作为种子像素点,此例的相似性准则是邻近点的灰度级与物体的平均灰度级的差小于 2。图中被接受的点和起始点均用下划线标出,图 11-5(b)是区域生长的第一步所得到的生长区域;图 11-5(c)是第二步

接受的邻近点，其种子像素点的灰度值是图 11-5(b)中所得到的生长区域的灰度平均值；图 11-5(d)是从像素点 6 开始生成的结果。

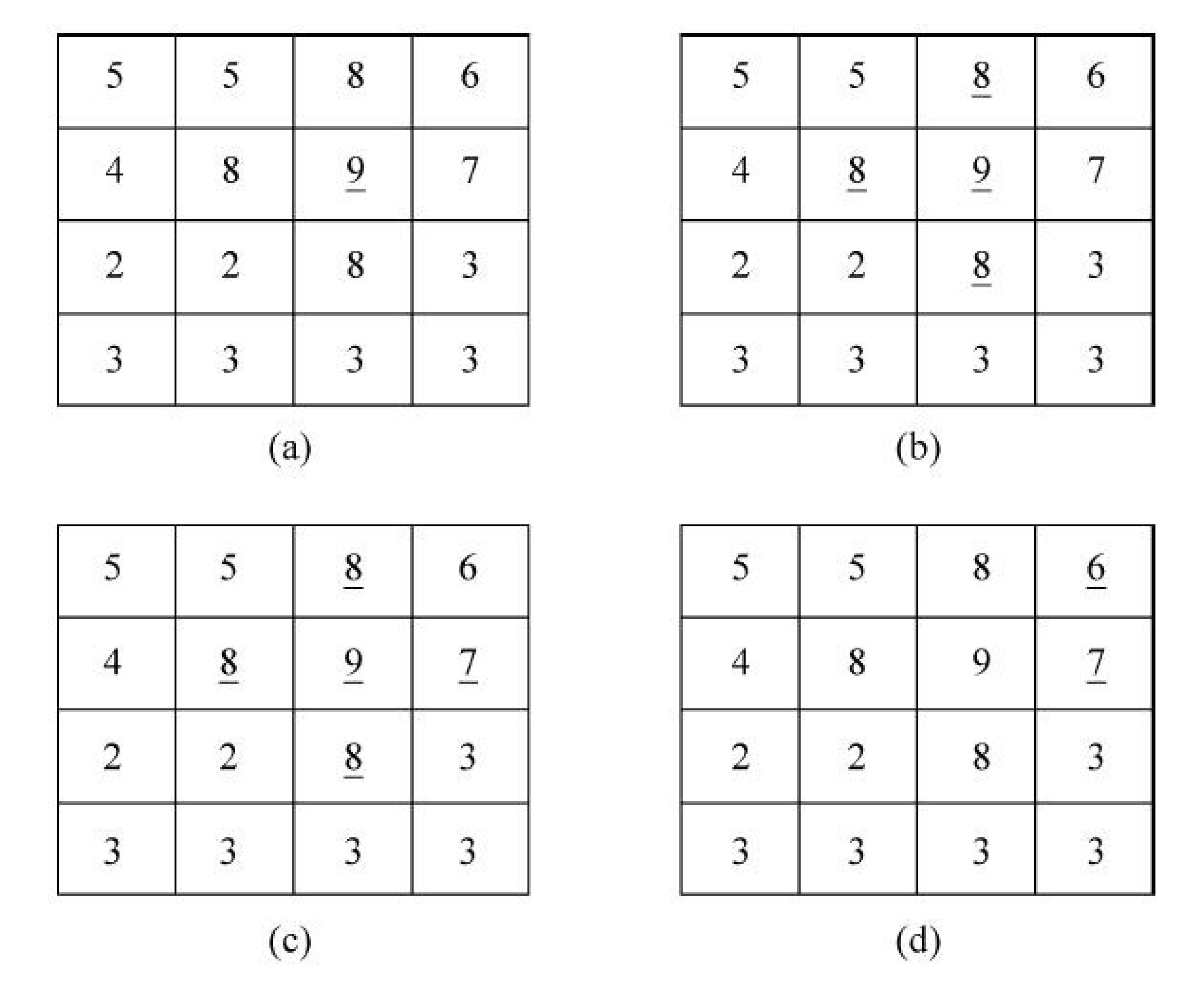


图 11-5 区域生长示例

当生成任意物体时，接收准则可以结构为基础，而不是以灰度级或对比度为基础。为了将候选的小群点包含在物体中，可以对这些小群点进行检测，而不是检测单个点，如果它们的结构与物体的结构足够相似时就接受它们。

实际应用中，需要注意如下问题：

(1) 确定图像中待分割区域的数目并确定每一区域的种子点，种子点一般可以根据具体的问题和专家经验进行选取，并没有一个统一的准则。

(2) 相似性度量准则的确定。度量准则的确定是区域生长的关键，既可以选取灰度级为度量准则，也可以选取图像的结构、纹理等非灰度级准则，其选取一般依赖于具体的分割目的。

(3) 区域生长法的优点是简单方便，但由于仅考虑了从一个像素点到另一个像素点的特性是否相似，因此对于有噪声或某些复杂的图像，应用该方法可能会引起一些不希望的区域出现。

11.3.2 区域分裂与合并

区域生长是从图像的某个或者某些像素点出发，最后得到整个区域，进而实现目标提取。分裂与合并法某种意义上是区域生长的逆过程：它是从整个图像出发，不断分裂得到各个子区域，然后再将前景区域合并，实现目标提取。分裂合并的假设是对于一幅图像，前景区域由一些相互连通的像素组成，因此，如果把一幅图像分裂到像素级，那么就可以判定该像素是否为前景像素。当所有像素点或者子区域完成判断以后，把前景区域或者像素合并就可得到前景目标。

另一种方法是在开始时将图像分割为一系列任意不相交的区域，然后将它们进行聚合或拆分以满足条件，即分离与合并的区域分析法，又称为区域分离与聚合法。该方法的基本

原理是利用图像数据的金字塔或四叉树数据结构的层次概念，将图像分为一组任意不相交的区域，即可以从图像的这种金字塔或四叉树结构的任意中间层开始，根据给定的均匀性检测准则进行分裂和合并这些区域，逐步改善这些区域的性能，直到最后将图像分为数量最少的区域为止。

令 R 表示整幅图像区域并选择一个均匀性准则 H。对 R 进行分割的一种方法是反复将分割得到的结果图像再次分为 4 个区域，直到对任何区域 R_i，有 $H(R_i)=\text{True}$。可以从整幅图像开始。如果 $H(R)=\text{False}$，就将图像分割为 4 个区域。对任何区域如果 H 的值是 False，就将这 4 个区域的每个区域再次分为 4 个区域，如此不断继续下去。如图 11-6 所示，这种特殊的分割技术可以方便地用四叉树（每个非叶子节点有 4 个子树）形式表示，其中，树的根对应于整幅图像，每个节点对应于划分的子区域，这里，只有 R_4 进行了进一步的再分。

如果仅使用拆分，最后的分区可能会包含具有相同性质的相邻区域。这一不足可以通过在进行拆分的同时，也允许进行区域聚合来予以纠正，限制条件仅需要将相邻区域进行聚合，但聚合的不同区域中相互连接的像素必须满足均匀性准则 H，即只有在 $H(R_k \cup R_j)=\text{True}$ 时，两个相邻的区域 R_j 和 R_k 才可以聚合。

因此，分裂聚合过程可以总结为如下步骤：

(1) 确定均匀性准则 H，将原始图像构成如图 11-6 所示的四叉树结构。

(2) 对于任何区域 R_i，如果 $H(R_i)=\text{False}$，就将每个区域拆分为 4 个相连的区域。

(3) 将 $H(R_k \cup R_j)=\text{True}$ 的任意两个相邻区域 R_k 和 R_j 进行聚合。

(4) 当无法进行进一步聚合或拆分时终止。

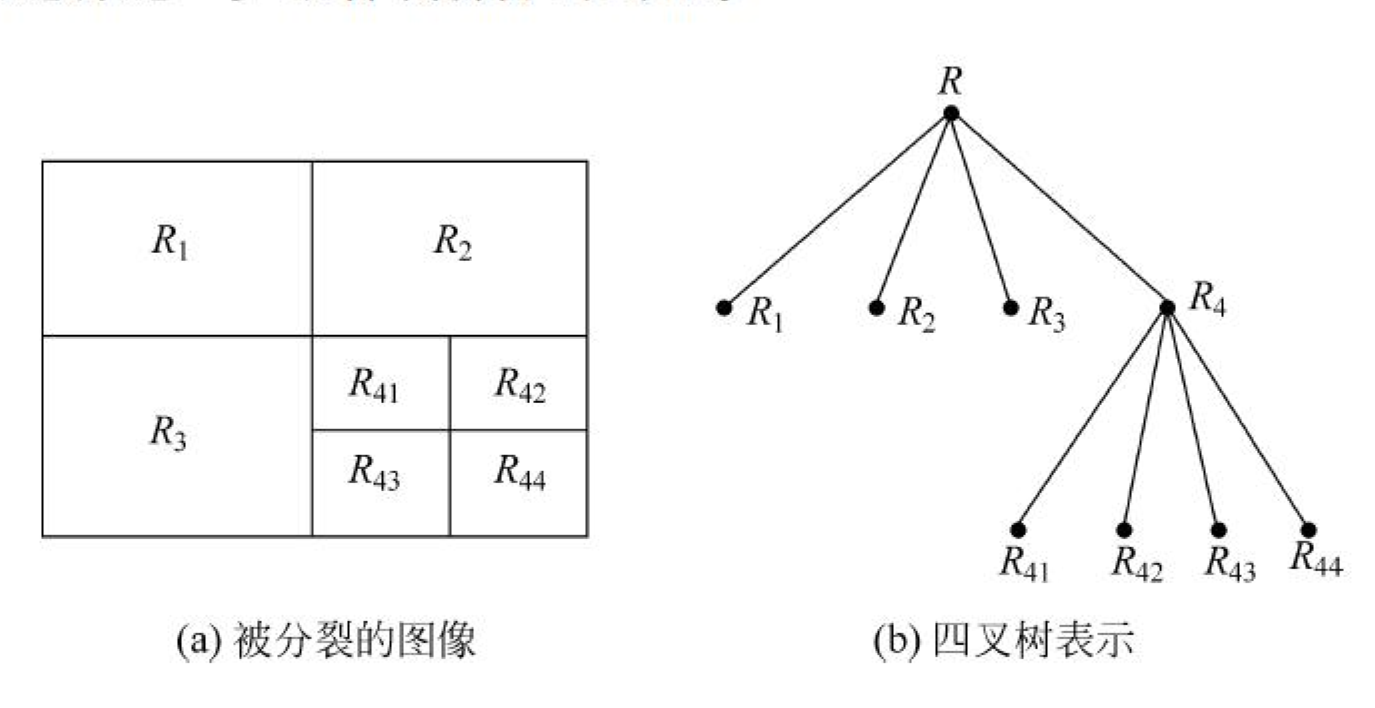


图 11-6 图像的分裂聚合四叉树结构

分裂与合并法的关键是分裂合并准则的设计。这种方法对复杂图像的分割效果较好，但算法较复杂，计算量大，分裂还可能破坏区域的边界。

可以对前面讲述的基本思想进行一些适当的变化。例如，一种变化方案是开始时将图像拆分为一组图像子块。然后对每个子块进一步进行上述拆分，但聚合操作开始时具有一定的限制，即只能将 4 块并为一组。这 4 块是四叉树表示法中节点的后代且都满足均匀性准则 H。当不能再进行此类聚合时，这个过程终止于满足步骤 3 的最后的区域聚合。在这种情况下，聚合的区域可能会大小不同。这种方法的主要优点是对于拆分和聚合都使用同样的四叉树，直到聚合的最后一步。

11.4 边缘检测的基本原理

图像分割的另一种重要途径是通过边缘检测，即检测灰度级或者一幅图像中结构具有突变的地方，表明一个区域的终结，也是另一个区域开始的地方。这种不连续性称为边缘。图像内不同的物体灰度不同，一般有明显的边缘，利用此特征可以分割图像。

图像内边缘处像素的灰度值不连续，这种不连续性可通过求导数检测。对于阶跃类型的边缘，其位置对应一阶导数的极值点，对应二阶导数的过零点。因此常用微分算子进行边缘检测。常用的一阶微分算子有 Robert 算子和 Sobel 算子，二阶微分算子有 Laplace 算子和 Kirsh 算子等。实际应用时各种微分算子常用模板来实现，即微分运算是利用模板和图像卷积实现的。这些算子对噪声敏感，只适合于噪声较小不太复杂的图像。

由于边缘和噪声都是灰度不连续点，从频域的概念分析都是高频分量，直接使用微分可能容易受噪声的影响。为了降低噪声的影响，用微分算子检测边缘前应对图像进行平滑滤波。LoG 算子和 Canny 算子是具有平滑功能的二阶和一阶微分算子，边缘检测效果较好。其中 LoG 算子是采用 Laplace 算子求高斯函数的二阶导数，Canny 算子是高斯函数的一阶导数，它在噪声抑制和边缘检测之间取得了较好的平衡。

11.4.1 边缘与边缘信号

物体的边缘无论是对人类的视觉系统还是对数字图像处理技术都具有非常重要的意义，它是图像的基本特征。根据 Marr 的视觉计算理论框架，提取二维图像物体上的边缘、角点、纹理等基本特征是整个系统框架的第一步，这些特征所组成的图称为基元图。边缘中包含图像物体有价值的边界信息，这些信息可以用于图像分析、滤波以及目标识别，并且通过边缘检测可以极大地降低后续图像分析处理的数据量。边缘存在于目标与背景、目标与目标、区域与区域、基元与基元之间。如图 11-7 所示，仅仅根据一幅图像的边缘点，就能识别出三维物体，可见边缘点确实包含了图像中的大量有用信息。



图 11-7 物体边缘的作用

11.4.2 边缘的类型

尽管边缘在数字图像处理和分析中具有重要作用，但是到目前为止，还没有关于边缘的被广泛接受和认可的精确的数学定义。一方面是因为图像的内容非常复杂，很难用纯数学的方法进行描述，另一方面则是因为人类对本身感知目标边界的高层视觉机理的认识还处于完善之中。

目前，具有对边缘的描述性定义，即两个具有不同灰度的均匀图像区域的边界，即边界反映局部的灰度变化。局部边缘是图像中局部灰度级以简单(即单调)的方式作极快变换的小区域。这种局部变化可用一定窗口运算的边缘检测算子来检测。边缘的描述包含以下几个方面：

(1) 边缘法线方向——在某点灰度变化最剧烈的方向，与边缘方向垂直。

(2) 边缘方向——与边缘法线方向垂直，是目标边界的切线方向。

（3）边缘强度——沿边缘法线方向图像局部的变化强度的量度。

一般认为沿边缘方向的灰度变化比较平缓，而边缘法线方向的灰度变化比较剧烈。图像上的边缘点可能对应不同的物理意义。如图11-8所示，图像上的边缘点包括以下几种情况。

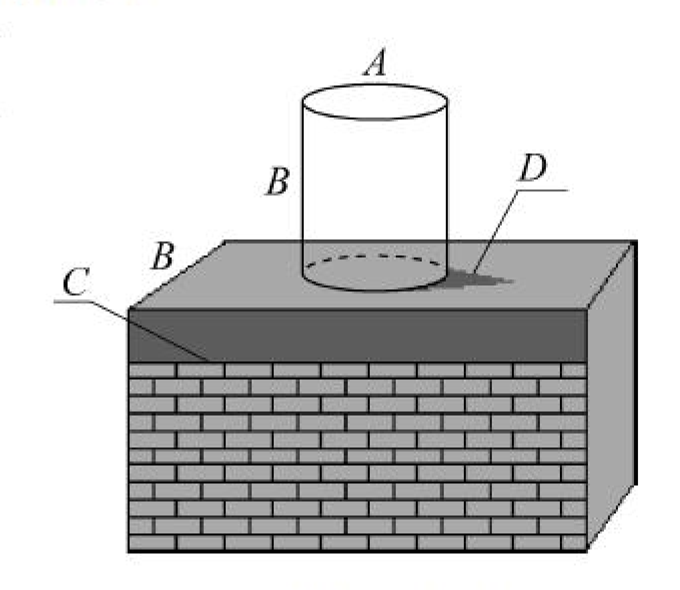


图11-8 图像中的各类边缘

（1）空间曲面上的不连续点。如标为 A 的边缘线，这些边缘线为两个不同曲面或平面的交线，该点处物体表面的法线方向不连续，在 A 类边缘线的两边，图像的灰度值有明显的不同。

（2）物体与背景的分界线。图中标注为 B 的边缘线是物体与背景的分界线，如图中圆柱上有两条 B 类边缘线，这类边缘线一般称为外轮廓线。在 B 类边缘点上，三维物体表面的法线方向是连续的，出现边缘点是由于从一定视角看物体时，B 类边界点是物体与背景的交界处。由于物体与背景在光照条件与材料反射系数等方面存在较大差异，因此在 B 类边缘两侧，图像的灰度也有很大差异。图中标注 B 的边缘是物体与背景的交界处，因此，边缘 B 是物体上表面法线的不连续处。但边缘 B 引起了两侧灰度的跃变。

（3）不同材料组成的边缘线。图中标注为 C 的边缘线是由不同材料或相同材料不同颜色所产生的。图中桌面由两种不同材料构成，由于它们对光的反射系数不同，使边缘线 C 的两侧灰度具有明显差别。

（4）阴影引起的边缘。图中标注为 D 的边缘线是由于阴影引起的边缘。由于物体表面某一部分被另一物体遮挡，使它得不到光源的照射或照射不充分，从而引起边缘点两侧灰度值有较大的差异。

11.4.3 典型边缘信号

尽管图像边缘点产生的原因各不相同，但它们都是图像上灰度不连续或灰度急剧变化的点，虽然这么说有些模糊，因此需要定义一个相对准确的边缘数学模型。根据信号理论，基本的灰度变化可以抽象为阶跃形或者脉冲形等形式，如图11-9(a)所示是一种阶跃信号，这时可以认为 A 点处为边缘点。在实际图像中，光照信号不可能有完全理想的突变，而可能是如图11-9(b)所示的逐渐增大的渐变信号，对图11-9(b)中所示 A、B、C 三点，一般定义 B 点为边缘点。在图11-9(c)中，如果台阶比较窄，即可以认为 B 点为边缘点，也可以认为该信号有两个边缘点 A 与 C。这些只是理想的情况，考虑到图像是二维的，而且往往图像上叠加有噪声，因此实际图像的灰度变化情况要复杂得多。

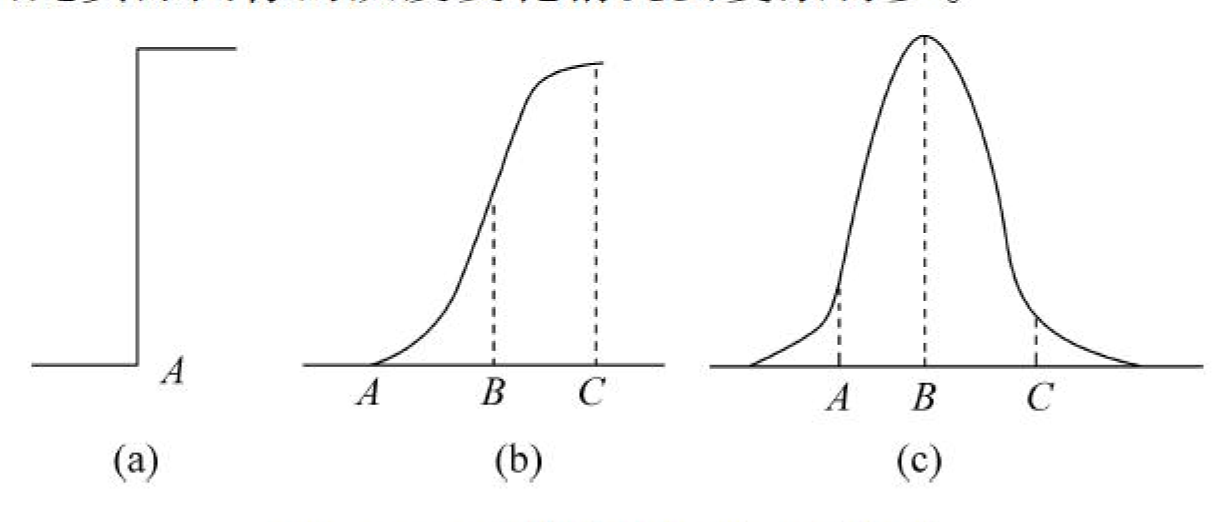


图11-9 图像中的各类边缘信号

11.5　边缘检测算子

图像中灰度值或色彩急剧变化之处，即为物体的边缘，在灰度值变换比较剧烈之处进行微分运算，就可以得出区别于其他处的较大数值，因此，可以利用各种微分运算进行边缘检测。边缘检测算子就是通过检查每个像素点的邻域并对其灰度变化进行量化来达到边界提取的目的，而且大部分的检测算子还可以确定边界变化的方向。前述章节中所讨论的各种基于微分的图像锐化算法都可以用于图像的边缘检测，也就是说，用各种锐化模板对图像进行卷积运算，便可检测出图像的边缘。

11.5.1　一阶微分算子

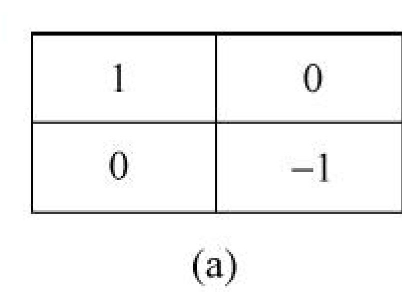
一阶微分算子主要有罗伯特(Robert)算子、索贝尔(Sobel)算子、普瑞维特(Prewitt)算子和凯西(Kirsch)算子，一般借助空域微分和卷积模板完成，对于数字图像，以差分近似微分来进行运算。

1. Robert 算子

罗伯特(Robert)边缘检测算子是一种利用局部差分方法寻找边缘的算子，于1963年提出，是第一个边缘检测算子，在边缘检测中得到了广泛应用。Robert 梯度算子所采用的是对角方向相邻两像素值之差，算子形式为

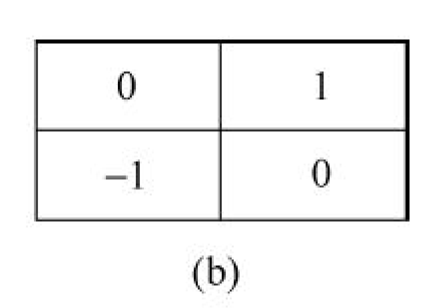
$$\begin{cases}\Delta_x f(x,y) = f(x,y) - f(x-1,y-1)\\ \Delta_y f(x,y) = f(x-1,y) - f(x,y-1)\end{cases} \tag{11-13}$$

上述算子对应的两个 2×2 模板如图 11-10 所示。实际应用中，图像中的每个像素点都用这两个模板进行卷积运算，为避免出现负值，在边缘检测时常取其绝对值。



1	0
0	−1

(a)



0	1
−1	0

(b)

图 11-10　Robert 算子模板

例 11-3：Robert 边缘检测算子示例

采用 Robert 算子对 lena 图像进行边缘检测，如图 11-11(a)所示为原始图像，图 11-11(b)为采用 Robert 算子对图 11-11(a)进行边缘检测的结果。



(a) 原始图像



(b) Robert算子检测结果

图 11-11　Robert 算子边缘检测结果

Robert 算子是利用局部差分算子寻找边缘的算子，采用对角线方向相邻两像素之差近似梯度幅值检测边缘，检测垂直边缘的效果好于斜向边缘。实验表明，Robert 对噪声敏感，无法抑制噪声的影响，算子定位精度与噪声有关，对具有陡峭边缘的低噪声图像有很好的效果。

2. Sobel 算子

索贝尔(Sobel)边缘算子所采用的算法是先进行加权平均，然后进行微分运算。算子的计算方法为

$$\begin{cases}\Delta_x f(x,y) = [f(x-1,y+1)+2f(x,y+1)+f(x+1,y+1)]- \\ \qquad [f(x-1,y-1)+2f(x,y-1)+f(x+1,y-1)] \\ \Delta_y f(x,y) = [f(x-1,y-1)+2f(x-1,y)+f(x-1,y+1)]- \\ \qquad [f(x+1,y-1)+2f(x+1,y)+f(x+1,y+1)]\end{cases} \tag{11-14}$$

Sobel 算子垂直方向和水平方向的模板如图 11-12 所示，前者可以检测出图像中的水平方向的边缘，后者则可以检测图像中垂直方向的边缘。实际应用中，每个像素点取两个模板卷积的最大值作为该像素点的输出值，运算结果是一幅边缘图像。

−1	−2	−1
0	0	0
1	2	1

(a)

−1	0	1
−2	0	2
−1	0	1

(b)

图 11-12　Sobel 算子模板

例 11-4：Sobel 边缘检测算子示例

本示例依然采用 lena 图像进行边缘检测，图 11-13 所示是用 Sobel 算子对图 11-11(a)进行边缘检测的结果。



图 11-13　Sobel 算子边缘检测结果

实验结果表明，Sobel 算子具有一定的噪声抑制能力，在检测渐变和噪声较多的图像时具有相对良好的效果，定位精度较好，检测阶跃边缘时可以得到至少两个像素的边缘宽度。

3. Prewitt 算子

普瑞维特(Prewitt)边缘检测算子是一种利用局部差分平均方法寻找边缘的算子，它体现了 3 对像素点像素值之差的平均概念，算子形式为

$$\begin{cases}\Delta_x f(x,y) = [f(x+1,y+1)+f(x,y+1)+f(x-1,y+1)]- \\ \qquad [f(x+1,y-1)+f(x,y-1)+f(x-1,y-1)] \\ \Delta_y f(x,y) = [f(x-1,y-1)+f(x-1,y)+f(x-1,y+1)]- \\ \qquad [f(x+1,y-1)+f(x+1,y)+f(x+1,y+1)]\end{cases} \tag{11-15}$$

Prewitt 边缘检测算子的两个模板如图 11-14 所示，它的使用方法同索贝尔算子一样，得到的结果也是一幅边缘图像。

−1	−1	−1
0	0	0
1	1	1

(a)

1	0	−1
1	0	−1
1	0	−1

(b)

图 11-14　Prewitt 算子模板

例 11-5：Prewitt 边缘检测算子示例

对 Prewitt 算子依然采用 lena 图像进行实验，图 11-15 所示是应用 Prewitt 算子对图 11-11(a)进行边缘检测的结果。



图 11-15　Prewitt 算子边缘检测结果

Prewitt 算子也是利用一阶微分运算边缘检测，其原理是在图像空间利用两个方向模板与图像进行邻域卷积来完成的，这两个方向模板一个检测水平边缘，一个检测垂直边缘。本实验表明，Prewitt 算子对噪声具有抑制作用，可去掉部分伪边缘，对于灰度渐变和噪声较多的图像具有相对好的效果。

4. Kirsch 算子

凯西(Kirsch)边缘检测算子需要求出 $f(x,y)$8 个方向的平均差分的最大值，其 8 个方向模板如图 11-16 所示，该算子可以检测出边缘的方向性信息，并能较好地抑制边缘检测的噪声。

5	5	5
−3	0	−3
−3	−3	−3

(a)

−3	5	5
−3	0	5
−3	−3	−3

(b)

−3	−3	5
−3	0	5
−3	−3	5

(c)

−3	−3	−3
−3	0	5
−3	5	5

(d)

−3	−3	−3
−3	0	−3
5	5	5

(e)

−3	−3	−3
5	0	−3
5	5	−3

(f)

5	−3	−3
5	0	−3
5	−3	−3

(g)

5	5	−3
5	0	−3
−3	−3	−3

(h)

图 11-16　Kirsch 算子模板

例 11-6：Kirsch 边缘检测算子示例

依然采用 lena 图像进行实验，图 11-17 所示是应用 Kirsch 算子对图 11-11(a)进行边缘检测的结果。

Kirsch 算子是 R. Kirsch 提出来的一种边缘检测算法，它采用 8 个 3×3 的模板对图像进行卷积，这 8 个模板代表 8 个方向，并取最大值作为图像的边缘输出。实验表明，该算子在保持细节和抗噪声方面都有较好的效果，比较适合于检测灰度渐变和噪声较多的图像。

图 11-17　Kirsch 算子边缘检测结果

11.5.2　二阶微分算子

二阶微分算子主要包括 Laplace 边缘检测算子和 Marr 边缘检测算子。

1. Laplace 算子

拉普拉斯(Laplace)边缘检测算子是二阶导数算子，与一阶差分算子相比，Laplace 算子能使噪声成分得到加强，对噪声更敏感。该算子是一种各向同性算子，在只关心边缘的位置而不考虑其周围的像素灰度差值时比较合适。其算子的形式为

$$\begin{cases} \Delta^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \\ \Delta^2 f(x,y) = f(x-1,y-1) + f(x,y-1) + f(x+1,y-1) + f(x-1,y) + \\ \qquad f(x+1,y) + f(x-1,y+1) + f(x,y+1) + f(x+1,y+1) - 8f(x,y) \end{cases} \tag{11-16}$$

拉普拉斯边缘检测算子的模板如图 11-18 所示，模板的基本特征是中心位置的系数为正，其余位置的系数为负，且模板的系数之和为零，通过与上述三个算子的比较，可以看出该算子不能检测出边缘的方向性信息，因此，它很少直接用于边缘检测。

0	−1	0
−1	4	−1
0	−1	0

(a)

−1	−1	−1
−1	8	−1
−1	−1	−1

(b)

图 11-18　Laplace 算子模板



图 11-19　Laplace 算子边缘检测结果

例 11-7：Laplace 边缘检测算子示例

采用 lena 图像进行 Laplace 算子实验，如图 11-19 所示是 Laplace 算子对图 11-11(a)进行边缘检测的结果。

实验结果表明，Laplace 算子对阶跃边缘定位较准确，该算子对孤立像素的响应比对边缘或线的响应更强烈，由于二阶微分对噪声的放大作用，对噪声很敏感，可能丢失边缘、产生边缘不连续。对于存在一定强度噪声的图像，使用 Laplace 算子检测边缘之前，需要先进行低通滤波。所以，通常需将该算子和平滑算子结合起来，生成一个新的模板。

2．Marr 边缘检测算子

由于微分运算对噪声具有一定的放大作用，因此，
对于存在较大噪声的图像，梯度算子和拉普拉斯算子进行边缘检测存在较大的误差。一种改进的方法是对噪声较大的图像进行噪声抑制，即先对图像进行平滑，然后再进行求微分运算，或者先对图像进行局部线性拟合，然后以光滑函数的导数代替直接的数值导数。

1）基本原理

虽然边缘检测的基本思想很简单，但在实际实现时却存在很大困难，其根本原因是，实际图像信号都存在有一定的噪声。以一维信号为例，如图 11-20(a)所示为一种理想信号，若存在噪声，则实际信号如图 11-20(b)所示。若采用前述方法，即一阶导数局部最大值或二阶导数过零点的方法检测边缘点，由于噪声一般是高频信号，在噪声的前沿或后沿，噪声信号的导数一般要高于边缘点处信号的导数，则检测出来的可能是噪声引起的假边缘点。解决这一问题的办法是先对信号进行平滑滤波，以滤去噪声。

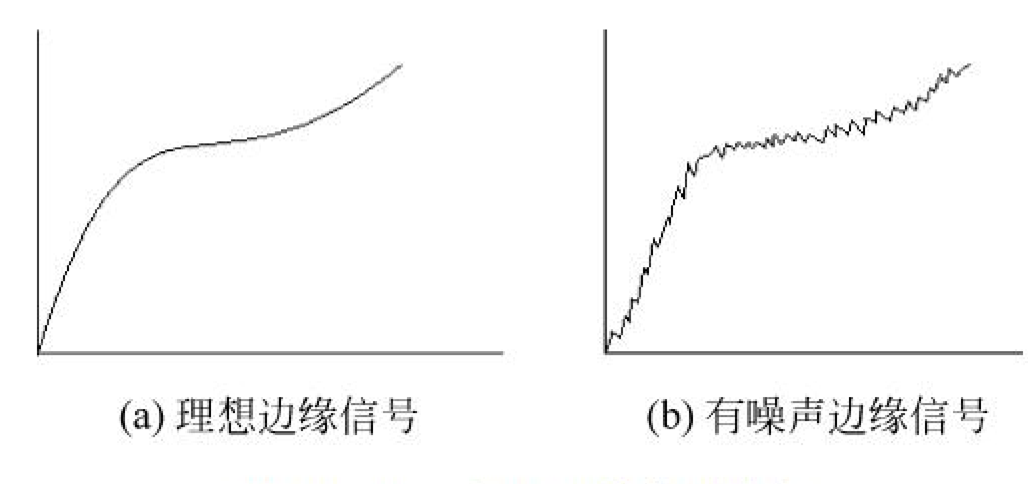

(a) 理想边缘信号　　(b) 有噪声边缘信号

图 11-20　实际边缘信号分析

设平滑滤波器的冲激响应函数以 $h(x)$表示，先对信号滤波，滤波后的信号为

$$g(x) = f(x) * h(x) \tag{11-17}$$

然后再对 $g(x)$进行求一阶或二阶导数运算以检测边缘点。因此有

$$\begin{aligned} g'(x) &= \frac{\mathrm{d}[f(x) * h(x)]}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}x}\int_{-\infty}^{\infty} f(t)h(x-t)\mathrm{d}t \\ &= \int f(t)h'(x-t)\mathrm{d}t = f(x) * h'(x) \end{aligned} \tag{11-18}$$

式中，$h'(x)$称为一阶微分滤波器。

上式表明，可以将先平滑后微分的两个步骤进行合并。因此，边缘检测的基本方法就是设计平滑滤波器 $h(x)$，检测 $f(x) * h'(x)$的局部最大值或二阶导数的过零点。

一般要求平滑滤波器 $h(x)$应满足以下条件：

(1) 当$|x|\to\infty$，$h(x)\to 0$，$h(x)$为偶函数。

(2) $\int_{-\infty}^{\infty} h(x)\mathrm{d}x = 1$，以保证信号经平滑滤波器 $h(x)$滤波后，其均值不变。

(3) $h(x)$一阶及两阶可微。

常用的平滑滤波器为高斯(Gauss)函数，即

$$h(x) = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{x^2}{2\sigma^2}} \tag{11-19}$$

式中，σ 为高斯函数的方差，称为高斯分布的空间尺度因子。若 σ 小，则表明函数“集中”，即仅在一个很小的局部范围内平滑；随着 σ 的增大，平滑的范围也相应增大；但 σ 太

大，虽然噪声被平滑了，但边缘点处同样也是信号变换较大的区域，于是边缘点可能也被平滑掉了，这一点是边缘检测时应引起注意的问题。

对上式求导可得出高斯函数的一阶和二阶导数为

$$h'(x) = -\frac{x}{\sqrt{2\pi}\sigma^3}e^{-\frac{x^2}{2\sigma^2}} \tag{11-20}$$

$$h''(x) = \left(\frac{x^2}{\sigma^2} - 1\right)\frac{1}{\sqrt{2\pi}\sigma^3}e^{-\frac{x^2}{2\sigma^2}} \tag{11-21}$$

2）Marr 边缘检测算法

Marr 边缘检测算子是一种用得较多的先平滑然后求导数的方法，先对图像做高斯滤波，然后再求其拉普拉斯二阶导数，通过检测滤波结果的零交叉(zero crossings)可以获得图像或物体的边缘。对于二维图像信号，先用下述高斯函数进行平滑：

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{11-22}$$

$G(x,y,\sigma)$是一个圆对称函数，其平滑的作用可通过 σ 进行控制。由于对图像进行线性平滑，在数学上是卷积运算，设 $f(x,y)$为平滑前的图像，$g(x,y)$为平滑后的图像，因此可得

$$g(x,y) = G(x,y,\sigma) * f(x,y) \tag{11-23}$$

由于边缘点是图像中灰度值变化剧烈的地方，这种图像灰度的突变将在一阶导数中出现一个峰，或等价于二阶导数中产生一个零交叉点。由于沿梯度方向的二阶导数是非线性的，计算较为复杂。于是 Marr 提出用如下拉普拉斯算子来替代(即以上式的零交叉点作为边缘点)：

$$\nabla^2 g(x,y) = \nabla^2(G(x,y,\sigma) * f(x,y)) = (\nabla^2 G(x,y,\sigma)) * f(x,y) \tag{11-24}$$

其中，上式中的$\nabla^2 G$ 称为 LoG(Laplace of Gaussian)算子，即高斯—拉普拉斯(LoG)算子，或高斯—拉普拉斯滤波器。进一步可得

$$\nabla^2 G(x,y,\sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} = \frac{1}{\pi\sigma^4}\left(\frac{x^2+y^2}{2\sigma^2} - 1\right)e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{11-25}$$

LoG 算子是由 Marr 提出的边缘检测算子，因此有时又被称为 Marr 算子。如图 11-21 所示是常用的 5×5 的 LoG 算子模板。

模板中各像素到中心点的距离与位置加权系数的关系可用图 11-22 所示曲线表示。若将图 11-22 绕 y 轴作旋转一周后，LoG 算子很像一顶墨西哥草帽，因此称为墨西哥草帽滤波器。

−2	−4	−4	−4	−2
−4	0	8	0	−4
−4	8	24	8	−4
−4	0	8	0	−4
−2	−4	−4	−4	−2

图 11-21　5×5 的 LoG 算子模板

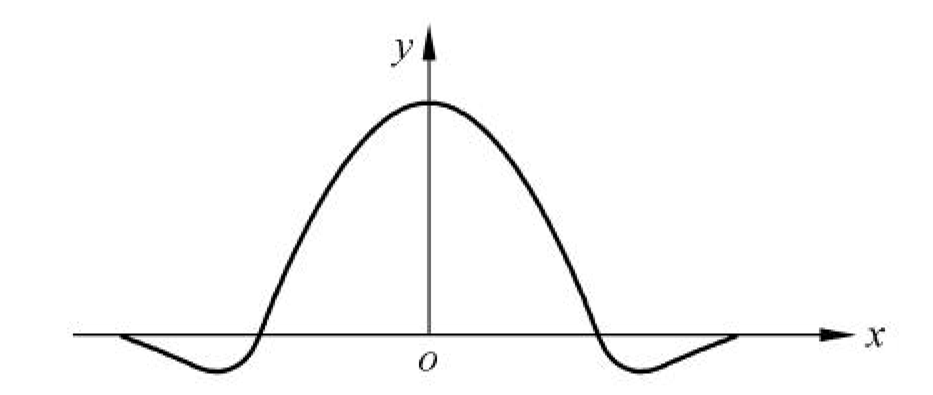


图 11-22　LoG 算子模板中心点的距离与位置加权系数关系曲线

LoG 算子将高斯平滑滤波器和拉普拉斯锐化滤波器结合起来，先平滑除去噪声，然后

再进行边缘检测，因此可以获得更好的效果。

例 11-8：LoG 边缘检测算子示例

如图 11-23 所示是用 5×5 的 LoG 算子对图 11-11(a) 所示 lena 图像进行边缘检测的结果。

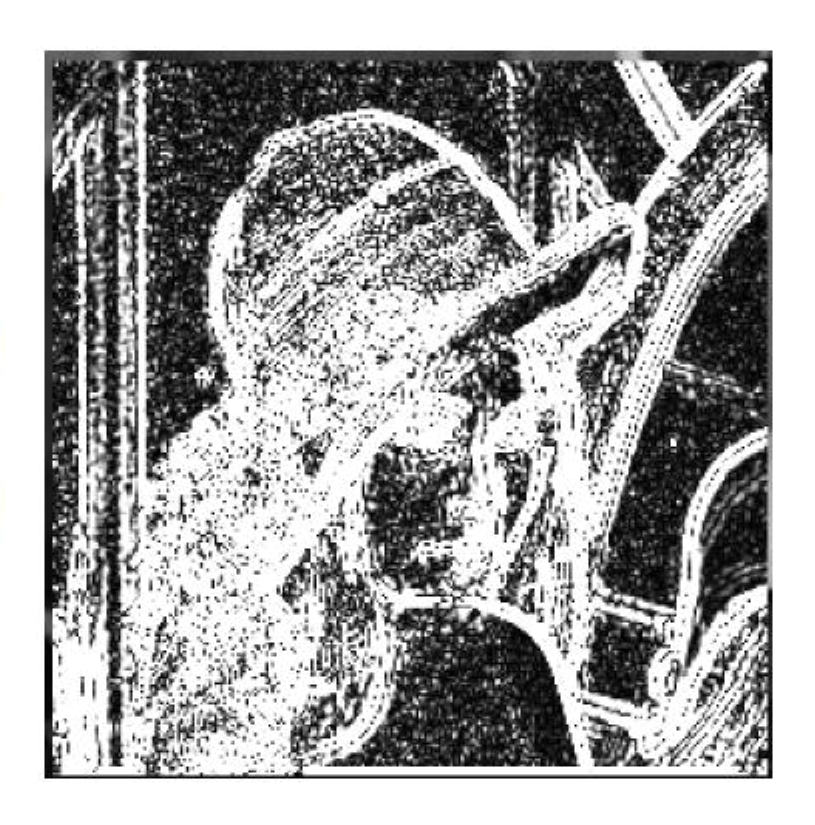


图 11-23　LoG 算子边缘检测结果

LoG 算子在 (x,y) 空间具有以原点为中心旋转的对称性，LoG 滤波器具有如下两个显著的特点：

(1) 该滤波器中的高斯函数部分对图像具有平滑作用，可有效地消除尺度远小于高斯分布因子 σ 的噪声信号。

(2) 高斯函数在空域和频域内都具有平滑作用。

(3) 该滤波器采用拉普拉斯算子 ∇^2 可以减少计算量。

11.5.3　Canny 边缘检测

Canny 边缘检测算子是 John F. Canny 于 1986 年提出的一个多级边缘检测算法。虽然 Canny 算法年代久远，但可以说它是边缘检测的一种标准算法，而且在图像研究中广泛使用。该算子边缘检测性能优越，适应性较广，至 2017 年 12 月，Canny 发表的该篇论文已被引用 22 000 余次。

1. 基本原理

Canny 创立了边缘检测计算理论，提出了如下最优边缘准则：

(1) 最优检测。边缘算法能够尽可能多地标识出图像中的实际边缘，漏检真实边缘的概率和误检非边缘的概率都尽可能小。

(2) 最优定位准则。检测到的边缘点的位置距离实际边缘点的位置最近，或者是由于噪声影响引起检测出的边缘偏离物体的真实边缘的程度最小。

(3) 检测点与边缘点一一对应。算子检测的边缘点与实际边缘点应该一一对应。

为了满足这些要求 Canny 使用了变分法，这是一种寻找优化特定功能函数的方法。最优检测使用 4 个指数函数项表示，但是它非常近似于高斯函数的一阶导数。

2. 边缘检测步骤

Canny 边缘检测算法可以分为如下 5 个步骤。

1) 图像平滑(去噪声)

边缘检测算法通常需要首先对原始图像进行铝箔，因此，该算法需要将原始数据与高斯模板进行卷积运算，消除图像噪声。这时，得到的图像与原始图像相比有些轻微的模糊。

2) 计算梯度幅值和方向

Canny 算法的基本思想是找寻一幅图像中灰度强度变化最强的位置，即指梯度方向，可以采用 Sobel 算子进行计算。

$$G=\sqrt{G_x^2+G_y^2},\qquad \theta=\arctan\left(\frac{G_x}{G_y}\right)$$

$$G_x=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix},\quad G_y=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

梯度角度 θ 范围从弧度 $-\pi$ 到 π，然后把它近似到 4 个方向，分别代表水平、垂直和两个对角线方向（0°，45°，90°，135°）。可以以 $\pm i\pi/8(i=1,3,5,7)$ 分割，落在每个区域的梯度角给一个特定值，代表 4 个方向之一。Sobel 算子得出来的边界通常非常粗，难以标定边界的真正位置。

3）非极大抑制

非最大值抑制是一种边缘细化方法。这一步的目的是将模糊的边界变得清晰（sharp）。通俗地讲，就是保留了每个像素点上梯度强度的极大值，而删掉其他的值。Canny 准则 3 要求，边缘只有一个精确的点宽度。非最大值抑制能帮助保留局部最大梯度而抑制所有其他梯度值。这意味着只保留了梯度变化中最锐利的位置。

对于每个像素点，进行如下操作：

（1）将其梯度方向近似为以下值中的一个（0°，45°，90°，135°，180°，225°，270°，315°）（即上下左右和 45°方向）。

（2）比较该像素点，和其梯度方向正负方向的像素点的梯度强度。

（3）如果该像素点梯度强度最大则保留，否则删除，即置为 0。

4）双阈值

经过非极大抑制后，图像中仍然有很多噪声点。Canny 算法中应用了一种叫双阈值的方法。即设定一个阈值上界和阈值下界（在 opencv 通常自己指定），图像中的像素点如果大于阈值上界，称为强边界（strong edge），则认为必然是边界；小于阈值下界，则认为必然不是边界，两者之间的称为弱边界（weak edge），认为是候选项，需进行进一步处理。

5）滞后边界跟踪

强边缘点可以认为是真的边缘。弱边缘点则可能是真的边缘，也可能是噪声或颜色变化引起的。为得到精确的边缘，后者引起的弱边缘点应该去掉。通常认为真实边缘引起的弱边缘点和强边缘点是连通的，而由噪声引起的弱边缘点则不会。所谓的滞后边界跟踪算法检查一个弱边缘点的 8 连通领域像素，只要有强边缘点存在，那么这个弱边缘点被认为是真实边缘保留下来，即 Canny 算法中，与强边界相连的弱边界认为是边界，其他的弱边界则被抑制。

例 11-9：Canny 边缘检测示例

如图 11-24(a)、(b)是 Canny 算子对彩色 lena 图像进行边缘检测的结果，图 11-24(c)、(d)是 Sobel 算子的边缘检测结果。Canny 算子相关参数：高斯半径 2，高阈值为 100，低阈值为 50；Sobel 算子阈值为 100。



(a) Canny算子梯度幅值



(b) Canny算子梯度二值图

图 11-24　Canny 算子边缘检测结果



(c) Sobel算子梯度幅值　　(d) Sobel算子梯度二值图

图 11-24(续)

该实验结果表明,Canny 边缘检测的效果是非常理想的,相比其他梯度算子,该算子较好地抑制了噪声引起的伪边缘,可以检测出真正的弱边缘,而且是经过细化的单边缘,易于后续处理。Canny 算子对于对比度较低的图像,通过合理选择参数,也能有很好的边缘检测效果。

需要指出的是,图像边缘检测算法较多,各种算子各有优缺点,具体应用中应根据算子的检测原理和研究的需要对算子进行合理的选择。

11.6 轮廓提取

数字图像轮廓提取的目的都是获得图像的外部轮廓特征,实现图像的轮廓跟踪与最终轮廓的获取,为进一步的图像分析做准备。

11.6.1 轮廓跟踪

顾名思义,轮廓跟踪就是通过顺序逐点找出边缘来跟踪目标边界。在识别图像中的目标时,往往需要对目标物体的边缘进行跟踪处理,即进行轮廓跟踪直到最后获得目标轮廓。若图像是二值图像或图像中不同的区域具有不同的像素值,但每个区域内的像素值是相同的,并具有 4 连通或 8 连通属性,则可以采用如下算法实现轮廓跟踪。

1. 寻找第一个边界点

首先按从上到下、从左到右的顺序扫描图像,寻找没有标记跟踪结束记号的第一个边界起始点 E_0,E_0 是具有最小行和列值的边界点。并定义一个扫描方向变量 dir 用于记录上一步中沿着前一个边界点到当前边界点的移动方向,dir 初始值如下:

(1) 对 4 连通区域取 dir=3,如图 11-25(a)所示。

(2) 对 8 连通区域取 dir=7,如图 11-25(b)所示。

2. 计算搜索方向

按逆时针方向搜索当前像素的 3×3 邻域,其起始搜索方向按如下方式设定:

(1) 对 4 连通区域取(dir+3)mod 4,如图 11-26(a)所示。

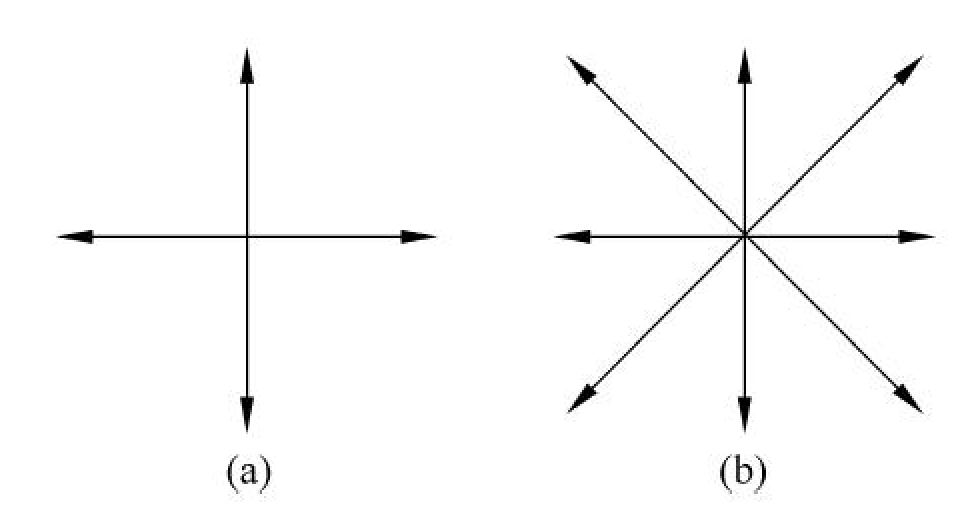
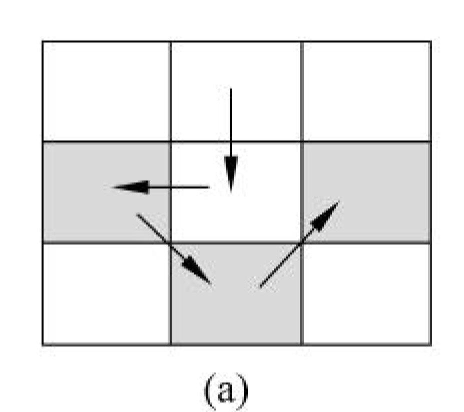
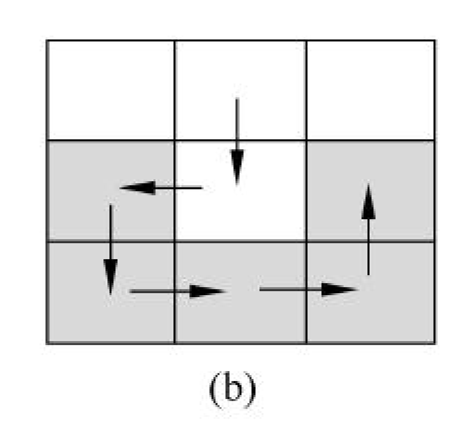


图 11-25 方向变量的初始化

（2）对 8 连通区域，若 dir 为奇数，取（dir＋7）mod 8，如图 11-26（b）所示；若 dir 为偶数，取（dir＋6）mod 8，如图 11-26（c）所示。





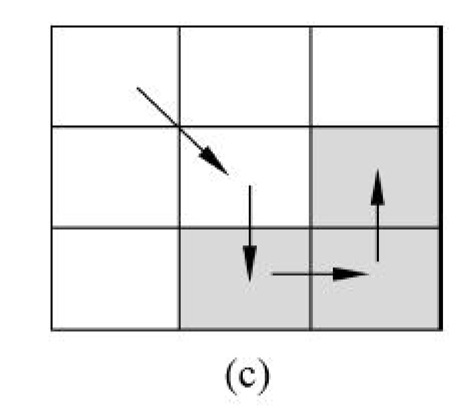


图 11-26 3×3 邻域起始搜索方向

在 3×3 邻域中，将搜索到的第一个与当前像素值相同的像素作为新的边界点 E_0，同时更新变量方向 dir 为新的值。

3. 找出所有边界点

如果 E_n 等于第二个边界点 E_1，且前一个边界点 E_{n-1} 等于第一个边界点 E_0，则停止搜索，结束跟踪，否则重复步骤 2 继续搜索。

4. 轮廓获取

由边界点 E_0、E_1、E_2、…、E_{n-1} 构成的边界就是所搜索的目标边界。

上述算法是图像轮廓跟踪的基本算法，如图 11-27（a）是原始图像，图 11-27（b）是应用上述算法进行轮廓跟踪的计算结果。根据图 11-27（b）可知，该算法只能跟踪目标图像的内边界，即边界包含在目标之内，而不能处理图像内的孔和洞。

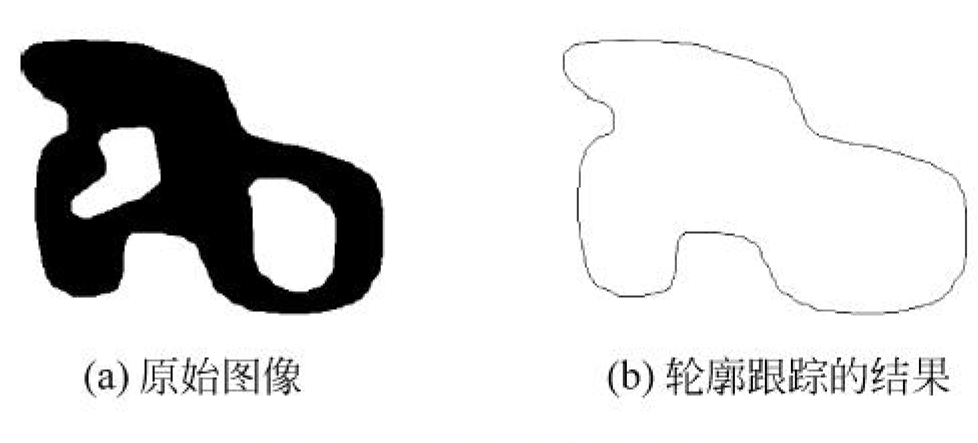


图 11-27 轮廓跟踪示例

11.6.2 二值图像轮廓

相对而言，二值图像轮廓提取的算法非常简单，就是采用逐点掏空目标物体的内部点的方法获取二值图像轮廓。如果原始图像中有一点为黑，且它的 8 个邻点的像素点都为黑时，说明该点是内部点，将该点删除，即置该像素点的灰度值为 255。对图像中所有像素点执行

同样的操作便可完成图像轮廓的提取。

11.7　图像匹配

图像匹配是指将不同时间、不同成像条件下对同一物体或场景获取的两幅或多幅图像在空间上对准，或根据已知模式到另一幅图像中寻找相应的模式。早期的图像匹配技术主要用于几何校正后的多波段遥感图像的匹配，借助于对互相关函数求极值来实现。

图像匹配可以应用在机器识别过程中，常需将不同传感器或同一传感器在不同时间、不同成像条件下对同一景物获取的两幅或多幅图像在空间上对准；在遥感图像处理中需将不同波段传感器对同一景物拍摄的多光谱图像按像点对应套准，然后根据像点的性质进行地物分类。如果利用在不同时间对同一地面拍摄的两幅照片，经套准后找出其中特征产生了变化的像点，便可分析图中哪些部分发生了变化。此外，图像匹配技术在对序列图像匹配、三维动态景物描述、物体的空间结构和运动参量计算等方面都发挥了重要的作用。

图像匹配包括模板匹配、直方图匹配、形状匹配等多种匹配方法，图像匹配技术已经在许多方面得到了广泛的应用。但对于三维空间，由于三维成像中有透视失真、运动遮挡、阴影和噪声干扰等不利因素，三维图像匹配依然是一个值得深入研究的课题。

11.7.1　模板匹配

模板匹配是指用一个较小的图像(即目标物形状模板)与原始图像进行比较，以确定在原始图像中是否存在与该模板相同或相似的区域。模板匹配通常可以检测出图像中的直线、曲线和边缘图像等，自 20 世纪 70 年代以来，模板匹配也得到了广泛的应用。例如，目标模板与侦查图像的匹配、目标模板与制导系统获得的图像进行匹配等。

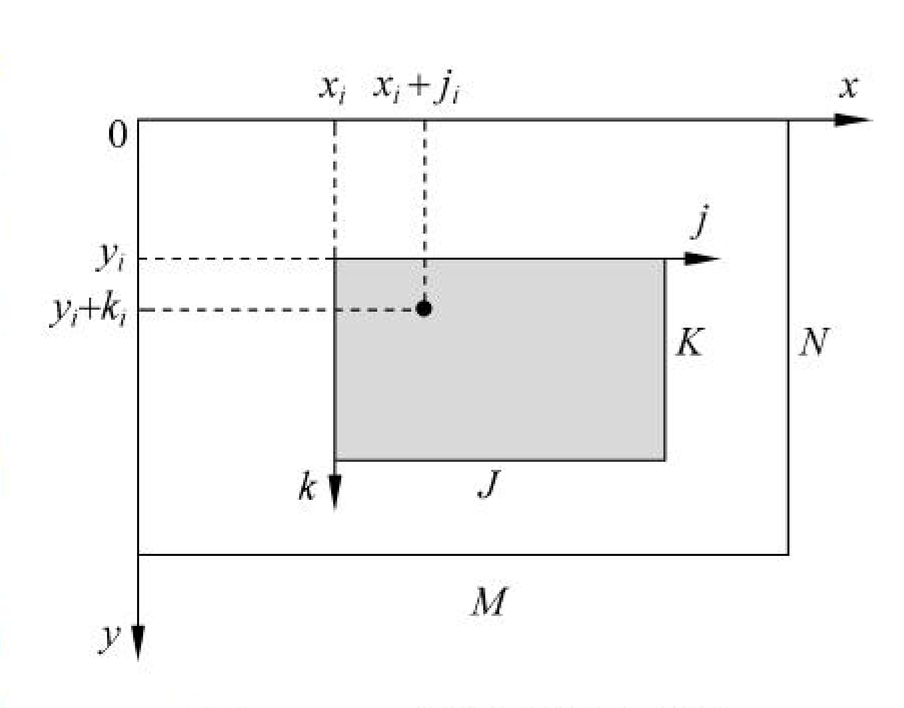


图 11-28　模板匹配示意图

模板匹配常用的一种测度方法为模板与原始图像对应区域的误差平方和。如图 11-28 所示为模板匹配的原理示意图。图中，设 $f(x,y)$ 表示 $M\times N$ 原图像，$t(j,k)$ 表示 $P\times Q(P\leqslant M, Q\leqslant N)$ 的模板图像，两者原点都在左上角。

定义模板匹配的误差平方和的测度为

$$D(x,y)=\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}[f(x+j,y+k)-t(j,k)]^2 \tag{11-26}$$

将上式展开可得

$$D(x,y)=\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}[f(x+j,y+k)]^2-2\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}f(x+j,y+k)t(j,k)+\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}[t(j,k)]^2$$


令

$$\mathrm{DS}(x,y)=\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}[f(x+j,y+k)]^2 \tag{11-27}$$

$$\mathrm{DST}(x,y)=2\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}f(x+j,y+k)t(j,k) \tag{11-28}$$

$$\mathrm{DT}(x,y)=\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}[t(j,k)]^2 \tag{11-29}$$

$\mathrm{DS}(x,y)$可定义为原图像中与模板对应区域的能量，它与像素位置(x,y)有关，但随像素位置(x,y)的变化，$\mathrm{DS}(x,y)$变化缓慢。$\mathrm{DST}(x,y)$可定义为模板与原图像对应区域的互相关性，它随像素位置(x,y)的变化而变化，当模板$r(j,k)$和原图像中对应区域相匹配时取得最大值。

$\mathrm{DT}(x,y)$称为模板的能量，根据表达式以及图11-28，该项与图像像素位置(x,y)无关，因此只需计算一次即可。

由于可定义$\mathrm{DT}(x,y)$与图像像素位置(x,y)无关，若$\mathrm{DS}(x,y)$近似为常数，则用$\mathrm{DST}(x,y)$便可进行图像匹配。以图11-28为例，根据代数极值理论，当模板与图像匹配时，$\mathrm{DST}(x,y)$取最大值。但假设$\mathrm{DS}(x,y)$为常数有时会产生误差，严重时将无法正确完成匹配，因此可以对上述测度进行适当改进，采用归一化互相关$R(x,y)$作为匹配误差平方和的测度，其形式为

$$R(x,y)=\frac{\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}f(x+j,y+k)t(j,k)}{\sqrt{\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}[f(x+j,y+k)]^2}\cdot\sqrt{\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1}[t(j,k)]^2}} \tag{11-30}$$

对任何一幅图像中的像素点(x,y)，根据式(11-30)都可以得出一个相应的$R(x,y)$值。当x和y变化时，$t(j,k)$在原图像区域中移动，从而可得出所有$R(x,y)$值。$R(x,y)$的最大值即指出了与$t(j,k)$匹配的最佳位置，若从该位置开始在原图像中取出与模板大小相同的子区域，就可以得到匹配图像，如图11-29所示。




图11-29　模板匹配结果

在应用模板匹配法进行图像匹配时，应注意以下3点：

(1) 归一化互相关法进行匹配，模板应在$(M-P+1)\times(N-Q+1)$个参考位置上进行相关计算，计算量非常大，必要时可以采用序贯相似性检测算法、幅度排序相关算法、FFT

相关算法、分层搜索序贯判决等方法对其进行改进，以提高运算速度。

(2) 模板匹配具有自身的局限性，主要表现在它只能进行平行移动，如原图像中的匹配目标发生旋转或大小变化，该算法无效。

(3) 若原图像中要匹配的目标只有部分可见，该算法也无法完成匹配。

11.7.2　直方图匹配

直方图匹配就是指利用图像的直方图及其特征量进行图像匹配。由于颜色是表示图像内容的一个重要方法，因此，作为图像检索中的一些算法就采用了基于颜色特征的搜索法，而且，目前已有多种借助颜色特征对图像进行检索的方法。现介绍几种典型的利用 R、G、B 颜色空间进行匹配的方法。

1. 直方图相交法

设 $H_Q(k)$ 和 $H_D(k)$ 分别为查询图像 Q 和数据库图像 D 的统计直方图，则两图像之间的匹配度 $R_{QD}(Q,D)$ 定义为

$$R_{QD}(Q,D)=\frac{\sum_{k=0}^{L-1}\min[H_Q(k),H_D(k)]}{\sum_{k=0}^{L-1}H_Q(K)} \tag{11-31}$$

2. 欧几里得距离法

显然，采用式(11-31)进行计算时，工作量非常大，为减少计算量，可采用直方图的均值来简略地表达颜色信息，对图像的 R、G、B 三个分量匹配，此时查询图像 Q 和数据库图像 D 之间的匹配度为

$$R_{QD}(Q,D)=\sqrt{(f_Q-f_D)^2}\sqrt{\sum_{R,G,B}(\mu_Q-\mu_D)^2} \tag{11-32}$$

式中，$f=[\mu_R,\mu_G,\mu_B]^{\mathrm{T}}$，$\mu_R,\mu_G,\mu_B$ 分别表示 R、G、B 直方图 0 阶矩。

3. 参考颜色法

由于直方图相交法计算量太大，欧几里得距离法太简略，因此，一种可以采取的折中办法是将图像颜色用一组参考色表示，这组参考色应能覆盖视觉上可感受到的各种颜色，但参考色的数量要比原图像少，这样可计算简化的直方图。

这种情况下，若以 r_i 表示第 i 种颜色出现的频率，N 表示参考颜色表的尺寸，则查询图像 Q 和数据库图像 D 之间的匹配度定义为

$$R_{QD}(Q,D)=\sqrt{\sum_{i=1}^{N}W_i(r_{iQ}-r_{iD})^2} \tag{11-33}$$

式中

$$W_i=\begin{cases}r_{iQ} & r_{iQ}>0 \text{ 且 } r_{iD}>0\\ 1 & r_{iQ}r_{iD}=0\end{cases}$$

4. 中心矩法

灰度直方图的灰度均值即为其 0 阶矩。实际上，图像的高阶矩也可以应用，若以 M_{QR}^i、M_{QG}^i M_{QB}^i 分别表示查询图像 Q 的 R、G、B 分量直方图的 i($i\leqslant3$)阶中心矩；以 M_{DR}^i、M_{DG}^i、M_{DB}^i 分别表示数据库图像 D 的 R、G、B 分量直方图的 i($i\leqslant3$)阶中心矩，则它们之间的匹配

度定义为

$$R_{QD}(Q,D)=\sqrt{W_R\sum_{i=1}^{3}(M_{QR}^i-M_{DR}^i)^2+W_G\sum_{i=1}^{3}(M_{QG}^i-M_{DG}^i)^2+W_B\sum_{i=1}^{3}(M_{QB}^i-M_{DB}^i)^2} \tag{11-34}$$

式中，W_R、W_G、W_B 为加权系数。

以上4种方法中，后3种主要是从减少计算量的角度对直方图相交法进行适当简化以减少运算量。直方图相交法的另一个问题是，当图像中的特征并不能取遍所有取值时，统计直方图中可能会出现一些零值，由于零值的出现将给直方图的相交带来影响，因此，有可能导致利用直方图相交法求出的匹配度不能正确反映匹配图像间的颜色差别。

值得注意的是，由于直方图不能反映颜色的位置信息，因此有可能出现两幅图像内容完全不同，但直方图相似。所以，仅用简单的颜色直方图匹配可能造成识别错误。因此，有必要对直方图匹配进行适当改进，改进的基本思路是利用匹配过程中的位置信息。例如，A. Nagasaka和 Y. Tanaka 提出的方法是将图像划分为若干可以一定程度上反映位置信息的子块，分别对各子块进行匹配，从而提高直方图匹配法的适应性。

5. X^2 直方图匹配

对于灰度图像，X^2 直方图匹配度定义为

$$R_{QD}(Q,D)=\sum_{k=0}^{L-1}\frac{[H_Q(k)-H_D(k)]^2}{H_D^r(k)} \tag{11-35}$$

而对于 R、G、B 图像，X^2 直方图匹配度的计算公式则为

$$R_{QD}(Q,D)=\sum_{k=0}^{L-1}\left\{\frac{[H_Q^R(k)-H_D^R(k)]^2}{H_D^R(k)}+\frac{[H_Q^G(k)-H_D^G(k)]^2}{H_D^G(k)}+\frac{[H_Q^B(k)-H_D^B(k)]^2}{H_D^B(k)}\right\} \tag{11-36}$$

X^2 直方图匹配与模板匹配或颜色直方图匹配相比具有更好的识别率，对于识别镜头切换(abrupt scene change)具有良好的效果。

A. Nagasaka 和 Y. Tanaka 通过对灰度模板匹配、彩色模板匹配、灰度直方图匹配、颜色直方图匹配和 X^2 直方图匹配多种方案进行了比较研究。结果表明，采用图像分块并结合 X^2 直方图匹配在镜头切换识别方面具有很好的效果，但对镜头渐变识别的效果依然存在一定的问题。

11.7.3 形状匹配

形状是描述图像内容的一个重要特征，因此，可以利用形状进行匹配。但采用形状进行匹配应综合考虑如下三个问题。

(1) 形状常与目标特征相联系，与颜色特征相比，形状特征被认为是更高层次的图像特征。要获取目标的有关形状参数，一般应先对图像进行分割，所以形状特征会受图像分割结果精度的影响。

(2) 目标形状的描述是一个非常抽象和复杂的问题，目前还没有与人的感觉一致且被大多数人所接受的关于图像形状描述的确切数学定义。

(3) 从不同视角和方法获取的图像中目标形状可能会有很大差别，为准确进行形状匹配，应保证平移、尺度、旋转变换的不变性等。

目标轮廓一定程度上可以表示目标物体的形状。轮廓是由物体的一系列边界点所形成的，因此根据图像分析的相关理论可知，在较大尺度下一般能较可靠地消除对物体边界点的误检，但却对边界的定位易产生不准确；相反，在较小尺度下，虽然增加了边界点误检的比例，但对物体或区域真正边界点的定位却比较准确。因此，可以结合两者的优点，即可先在较大尺度下检测出真正的边界点，然后在较小尺度下对边界点进行较精确的定位。

目前，常用的形状匹配法主要有不变矩法、几何参数法、特征模表示法、边界方向直方图法、小波重要系数法以及我国学者杨翔英、章毓晋于1999年发表的小波轮廓表示法等许多方法。有着数学显微镜之称的小波变换与分析作为一种多尺度、多分辨率的分析工具，比较适合于对图像进行多尺度的边界检测。

11.8 边缘检测的MATLAB实现

11.8.1 程序代码

```
function EdgeDetection(action, varargin)
%基于差分运算的数字图像边缘检测软件主要函数说明
%InitializeEdgeDetection :检测程序相关功能初始化
%ComputeEdgeMap          :计算原始图像边缘
%SelectMethod            :选择边缘检测方法
%LoadNewImage            :读入被选择的图像
%UpdatesprThreshCtrl     :数据更新
%UpdateLOGSize -         :更新与LOG滤波器相关的数据
if nargin<1,
  action = 'InitializeEdgeDetection';
end;
feval(action,varargin{:});
return;
%%% 函数 InitializeEdgeDetection
function InitializeEdgeDetection()
h = findobj(allchild(0), 'tag', 'EdgeDetectionSoft');
if ~isempty(h)
  figure(h(1))
  return
end
screenD = get(0, 'ScreenDepth');
if screenD>8
  grayres = 256;
else
  grayres = 128;
end
EdgeDetectionFig = figure( ...
  'Name','基于差分运算的数字图像边缘检测软件', ...
  'NumberTitle','off', 'HandleVisibility', 'on', ...
  'tag', '基于差分运算的数字图像边缘检测软件', ...
  'Visible','off', 'Resize', 'off',...
  'BusyAction','Queue','Interruptible','off', ...
  'Color', [.8 .8 .8], ...
```

```
    'IntegerHandle', 'off', ...
    'DoubleBuffer', 'on', ...
    'Colormap', gray(grayres));
figpos = get(EdgeDetectionFig, 'position');
%调节图像窗口尺寸
figpos(3:4) = [560 420];
horizDecorations = 10;   % resize controls, etc.
vertDecorations = 45;    % title bar, etc.
screenSize = get(0,'ScreenSize');
if (screenSize(3) <= 1)
    % No display connected (apparently)
    screenSize(3:4) = [100000 100000];
end
if (((figpos(3) + horizDecorations) > screenSize(3)) | ...
            ((figpos(4) + vertDecorations) > screenSize(4)))
    %窗口尺寸太小
    delete(EdgeDetectionFig);
    error(['Screen resolution is too low ', ...
           '(or text fonts are too big) to run this edgesoft']);
end
dx = screenSize(3) - figpos(1) - figpos(3) - horizDecorations;
dy = screenSize(4) - figpos(2) - figpos(4) - vertDecorations;
if (dx < 0)
    figpos(1) = max(5,figpos(1) + dx);
end
if (dy < 0)
    figpos(2) = max(5,figpos(2) + dy);
end
set(EdgeDetectionFig, 'position', figpos);
rows = figpos(4); cols = figpos(3);
hs = (cols - 512) / 3;     %垂直间距
bot = rows - 2 * hs - 256;  %图像底部
%按钮和菜单参数
ifs = hs/2;
Std.Interruptible = 'off';
Std.BusyAction = 'queue';
%原始图像轴线
hdl.ImageAxes = axes(Std, ...
    'Units', 'Pixels', ...
    'Parent',EdgeDetectionFig,...
    'ydir', 'reverse', ...
    'XLim', [.5 256.5], ...
    'YLim', [.5 256.5],...
    'CLim', [0 255], ...
    'Position',[hs bot 256 256], ...
    'XTick',[],'YTick',[]);
set(get(hdl.ImageAxes, 'title'), 'string', '原始图像');
%边缘图像轴线
hdl.EdgeAxes = axes(Std, ...
    'Units', 'Pixels', ...
    'Parent',EdgeDetectionFig,...
```

```
  'ydir', 'reverse', ...
  'XLim', [.5 256.5], ...
  'YLim', [.5 256.5],...
  'CLim', [0 1], ...
  'Position',[cols - hs - 256 bot 256 256], ...
  'XTick',[],'YTick',[]);
set(get(hdl.EdgeAxes, 'title'), 'string', '边缘图像');
%原始图像
hdl.Image = image(Std, ...
  'CData', [], ...
  'CDataMapping', 'scaled', ...
  'Parent',hdl.ImageAxes,...
  'Xdata', [1 256],...
  'Ydata', [1 256],...
  'EraseMode', 'none');
%边缘图像
hdl.Edge = image(Std, ...
  'CData', [], ...
  'CDataMapping', 'scaled', ...
  'Parent',hdl.EdgeAxes,...
  'Xdata', [1 256],...
  'Ydata', [1 256],...
  'EraseMode', 'none');
%Background color for frames
bgcolor = [0.45 0.45 0.45];
fgcolor = [1 1 1]; % For text
%菜单框架
mfleft = hs;
mfbot = hs;
mfwid = (3 * cols/8) - 1.5 * hs;
mfht = bot - 2 * hs;
hdl.MenuFrame = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style', 'frame', ...
  'Units', 'pixels', ...
  'Position', [mfleft mfbot mfwid mfht], ...
  'BackgroundColor', bgcolor);
%图像下拉菜单
ipwid = mfwid - 2 * ifs;
ipht = 21;
ipleft = mfleft + ifs;
ipbot = mfbot + 1.7 * ifs + 2 * ipht;
hdl.ImgPop = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','popupmenu', ...
  'Units','pixels', ...
  'Position',[ipleft ipbot ipwid ipht], ...
  'Enable','on', ...
  'String','米粒|土星|钱币 1|电路|轮盘|飞船|钱币 2|玻璃', ...
  'Tag','ImagesPop',...
  'Callback','EdgeDetection(''LoadNewImage'')');
```

```
%图像菜单标签
uicontrol( Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','text', ...
  'Units','pixels', ...
  'Position',[ipleft ipbot + ipht ipwid 18], ...
  'Horiz','left', ...
  'Background',bgcolor, ...
  'Foreground',fgcolor, ...
  'String','选择原始图像:');
%算子选择菜单
hdl.Method = 'Sobel';
mpwid = ipwid;
mpht = ipht;
mpleft = ipleft;
mpbot = mfbot + 1.2 * ifs;
hdl.MethodPop = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','popupmenu', ...
  'Units','pixels', ...
  'Position',[mpleft mpbot mpwid mpht], ...
  'Enable','on', ...
  'String','Sobel|Prewitt|Roberts|Laplacian of Gaussian|Canny', ...
  'Tag','MethodPop',...
  'Callback','EdgeDetection(''SelectMethod'')');
%算子选择菜单标签文本
uicontrol( Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','text', ...
  'Units','pixels', ...
  'Position',[mpleft mpbot + mpht mpwid 18], ...
  'Horiz','left', ...
  'Background',bgcolor, ...
  'Foreground',fgcolor, ...
  'String','选择边缘检测算子:');
%框架参数
pfleft = (3 * cols/8) + 0.5 * hs;
pfbot = 1.5 * hs;
pfwid = (3 * cols/8) - hs;
pfht = bot - 2.5 * hs;
hdl.ParamFrame = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style', 'frame', ...
  'Units', 'pixels', ...
  'Position', [ pfleft pfbot pfwid pfht ], ...
  'BackgroundColor', bgcolor);
%域值控制文本标签
labelleft = pfleft + ifs;
labelwid = pfwid/2 - hs;
labelbot = pfbot + 2 * pfht/3;
hdl.sprThLbl = uicontrol(Std,...
```

```
    'Parent', EdgeDetectionFig, ...
    'Style','text', ...
    'Units','pixels', ...
    'Position',[labelleft labelbot labelwid 18], ...
    'Horiz','left', ...
    'String','域 值:', ...
    'BackgroundColor',bgcolor, ...
    'ForegroundColor',fgcolor);
hdl.Threshold = 0;
raleft = pfleft + pfwid/2 - hs/2;
rabot = pfbot + 2 * pfht/3 + hs/6;
rawid = pfwid/2;
raht = ipht;
hdl.RadioAutomatic = uicontrol(Std, ...
    'Parent', EdgeDetectionFig, ...
    'Style','radiobutton', ...
    'Units','pixels', ...
    'Position',[raleft rabot rawid raht], ...
    'String','自动匹配', ...
    'value',1,'Userdata',1, ...
    'Callback','EdgeDetection(''Radio'',''auto'')');
rmleft = pfleft + pfwid/2 - hs/2;
rmbot = pfbot + pfht/3 + hs/3;
rmwid = hs * 1.5;
rmht = ipht;
hdl.RadioManual = uicontrol(Std, ...
    'Parent', EdgeDetectionFig, ...
    'Style','radiobutton', ...
    'Units','pixels', ...
    'Position',[rmleft rmbot rmwid rmht], ...
    'String','', ...
    'value',0,'Userdata',0, ...
    'Callback','EdgeDetection(''Radio'',''manual'')');
thleft = rmleft + rmwid;
thwid = rawid - rmwid;
thbot = rmbot;
thht = rmht;

hdl.ThreshCtrl = uicontrol(Std, ...
    'Parent', EdgeDetectionFig, ...
    'Enable', 'off', ...
    'Style','edit', ...
    'Units','pixels', ...
    'Position',[thleft thbot thwid thht], ...
    'Horiz','right', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0',...
    'callback','EdgeDetection(''UpdateSprThresh'')');
%算子方向选择菜单
dpwid = pfwid/2;
```

```
dpht = ipht;
dpleft = pfleft + pfwid/2 - hs/2;
dpbot = pfbot + .4 * hs;
hdl.sprDirPop = uicontrol(Std, ...
   'Parent', EdgeDetectionFig, ...
   'Style','popupmenu', ...
   'Units','pixels', ...
   'Position',[dpleft dpbot dpwid dpht], ...
   'Enable','on', ...
   'String','水平垂直|水平方向|垂直方向', ...
   'Tag','DirectionPop',...
   'Callback','EdgeDetection(''UpdateDirectionality'')');
%算子方向选择文本标签
labelleft = pfleft + ifs;
labelwid = pfwid/2 - hs;
labelbot = dpbot;
hdl.sprDirLbl = uicontrol( Std, ...
   'Parent', EdgeDetectionFig, ...
   'Style','text', ...
   'Units','pixels', ...
   'Position',[labelleft labelbot labelwid 18], ...
   'Horiz','left', ...
   'Background',bgcolor, ...
   'Foreground',fgcolor, ...
   'String','算子方向:');
hdl.Directionality = 'both';
hdl.logSigmaCtrl = uicontrol(Std, ...
   'Parent', EdgeDetectionFig, ...
   'Style','edit', ...
   'Units','pixels', ...
   'Position',[dpleft dpbot dpwid dpht], ...
   'Horiz','right', ...
   'Background','white', ...
   'Foreground','black', ...
   'String','2', ...
   'Tag','DirectionPop',...
   'Visible', 'off', ...
   'Callback','EdgeDetection(''UpdateLOGSigma'')');
%Sigma 编辑框文本
hdl.logSigmaLbl = uicontrol( Std, ...
   'Parent', EdgeDetectionFig, ...
   'Style','text', ...
   'Units','pixels', ...
   'Position',[labelleft labelbot labelwid 18], ...
   'Horiz','left', ...
   'Background',bgcolor, ...
   'Foreground',fgcolor, ...
   'Visible', 'off', ...
   'String','Sigma:');
hdl.LogSigma = 2;
%状态条
```

```
colr = get(EdgeDetectionFig,'Color');
hdl.Status = uicontrol( Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','text', ...
  'Units','pixels', ...
  'Background', colr, ...
  'Foreground', [.8 0 0], ...
  'Position',[pfleft 2 pfwid 18], ...
  'Horiz','center', ...
  'Tag', 'Status', ...
  'String','Initializing EdgeDetectionSoft...');
%消息等有关按钮框
bfleft = (3 * cols/4) + .5 * hs;
bfbot = hs;
bfwid = (cols/4) - 1.5 * hs;
bfht = bot - 2 * hs;
hdl.ButtonFrame = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style', 'frame', ...
  'Units', 'pixels', ...
  'Position', [ bfleft bfbot bfwid bfht ], ...
  'BackgroundColor', bgcolor);
%运行程序按钮设计
btnwid = bfwid - 2 * ifs;
btnht = (bfht - 4 * ifs)/3;% 21
btnleft = bfleft + ifs;
btnbot = bfbot + bfht - ifs - btnht;
hdl.Apply = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','pushbutton', ...
  'Units','pixels', ...
  'Position',[btnleft btnbot btnwid btnht], ...
  'Enable','off', ...
  'String','运行程序', ...
  'Callback','EdgeDetection(''ComputeEdgeMap'')');
%消息按钮
btnbot = bfbot + bfht - 2 * ifs - 2 * btnht;
hdl.Help = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','pushbutton', ...
  'Units','pixels', ...
  'Position',[btnleft btnbot btnwid btnht], ...
  'Enable','off', ...
  'String','信息', ...
  'Callback','helpwin EdgeDetection');
%退出按钮
btnbot = bfbot + ifs;
hdl.Close = uicontrol(Std, ...
  'Parent', EdgeDetectionFig, ...
  'Style','pushbutton', ...
  'Units','pixels', ...
```

```
    'Position',[btnleft btnbot btnwid btnht], ...
    'Enable','off', ...
    'String','退出', ...
    'Callback','close(gcbf)');
set(EdgeDetectionFig, 'Userdata', hdl, 'Visible', 'on');
drawnow
LoadNewImage(EdgeDetectionFig);
drawnow
set(EdgeDetectionFig, 'HandleVisibility', 'Callback');
set([hdl.Apply hdl.Help hdl.Close] , 'Enable', 'on');
return

% 函数 ComputeEdgeMap
function ComputeEdgeMap(MyEdgeFig)
if nargin<1
  callb = 1;
  MyEdgeFig = gcbf;
else
  callb = 0;                % 初始化
end
set(MyEdgeFig,'Pointer','watch');
setstatus(MyEdgeFig, '计算边缘图像...');
hdl = get(MyEdgeFig,'Userdata');
img = getimage(hdl.Image);
autothresh = get(hdl.RadioAutomatic, 'Value');
switch hdl.Method
case {'Sobel','Roberts','Prewitt'}
  if autothresh
    [edgemap,thresh] = edge(img, hdl.Method, hdl.Directionality);
    setstatus(['域值为: 'num2str(thresh) '.']);
  else
    edgemap = edge(img, ...
      hdl.Method, hdl.Threshold, hdl.Directionality);
    setstatus(MyEdgeFig, '');
  end
case 'Laplacian of Gaussian'
  if autothresh
    [edgemap,thresh] = edge(img, 'log', [], hdl.LogSigma);
    setstatus(MyEdgeFig, ['域值为: 'num2str(thresh) '.']);
  else
    edgemap = edge(img, 'log', hdl.Threshold, hdl.LogSigma);
    setstatus(MyEdgeFig, '');
  end
case 'Canny'
  if autothresh
    [edgemap,thresh] = edge(img, 'canny', [], hdl.LogSigma);
    % statstr = sprintf('Thresholds are [%0.4f,0.4f].', thresh(1),thresh(2));
    setstatus(MyEdgeFig, ['高域值为: 'num2str(thresh(2)) '.']);
  else
    [edgemap,thresh] = edge(img, 'canny', hdl.Threshold, hdl.LogSigma);
    % statstr = sprintf('Thresholds are [%0.4f,0.4f].', thresh(1),thresh(2));
```

```
    setstatus(MyEdgeFig, '');
  end
otherwise
  error('EdgeDetection: Invalid edge detection method.');
end
set(hdl.Edge, 'CData', edgemap);
set(hdl.Apply, 'Enable', 'off');
set(MyEdgeFig,'Pointer','arrow');
drawnow
%函数 SelectMethod
function SelectMethod
MyEdgeFig = gcbf;
hdl = get(MyEdgeFig, 'userdata');
v = get(hdl.MethodPop,{'value','String'});
hdl.Method = deblank(v{2}(v{1},:));
switch hdl.Method
case {'Sobel','Prewitt'}
  ActivateSPRControls(MyEdgeFig);
  set(hdl.sprDirPop, 'Enable', 'on');
case 'Laplacian of Gaussian'
  ActivateLOGControls(MyEdgeFig);
  set(hdl.logSigmaCtrl, 'String', '2');
  hdl.LogSigma = 2;
case 'Canny'
  ActivateLOGControls(MyEdgeFig);
  set(hdl.logSigmaCtrl, 'String', '1');
  hdl.LogSigma = 1;
case 'Roberts'
  ActivateSPRControls(MyEdgeFig);
  set(hdl.sprDirPop, 'Enable', 'off', 'value', 1);
otherwise
  error('EdgeDetection: invalid method specifier.');
end
set(hdl.Apply, 'Enable', 'on');
set(MyEdgeFig, 'userdata', hdl);
%BlankOutEdgeMap(MyEdgeFig);
setstatus(MyEdgeFig, ['单击 ''运行程序按钮''计算图像边缘.']);

%函数 LoadNewImage
function LoadNewImage(MyEdgeFig)
if nargin<1
  callb = 1;
  MyEdgeFig = gcbf;
else
  callb = 0;
end
set(MyEdgeFig,'Pointer','watch');
hdl=get(MyEdgeFig,'Userdata');
v = get(hdl.ImgPop,{'value','String'});
name = deblank(v{2}(v{1},:));
drawnow
```

```
switch name
case '飞船',
  liftbody256 = [];
  load imdemos liftbody256
  img = liftbody256;
case '钱币 1',
  coins2 = [];
  load imdemos coins2
  img = coins2;
case '米粒',
  rice3 = [];
  load imdemos rice3
  img = rice3;
case '土星',
  saturn2 = [];
  load imdemos saturn2
  img = saturn2;
case '钱币 2',
  eight = [];
  load imdemos eight
  img = eight;
case '电路',
  circuit4 = [];
  load imdemos circuit4
  img = circuit4;
case '轮盘',
  vertigo2 = [];
  load imdemos vertigo2
  img = vertigo2;
case '玻璃',
  glass2 = [];
  load imdemos glass2
  img = glass2;
otherwise
  error('EdgeDetection: Unknown Image Option!');
end
set(hdl.Image, 'Cdata', img);
set(get(hdl.ImageAxes,'title'),'string',['原始' name '图像']);
set(MyEdgeFig,'Pointer','arrow');
if callb
  set(hdl.Apply, 'Enable', 'on');
end
drawnow

% 应用 LoG 算子时,采用自动域值计算
if ~strcmp(hdl.Method, 'Laplacian of Gaussian')
  if get(hdl.RadioAutomatic, 'Value') == 0
    Radio('auto',MyEdgeFig);
  end
end
```

```
ComputeEdgeMap(MyEdgeFig);
return;

% 函数 UpdateSprThresh
function UpdateSprThresh()
MyEdgeFig = gcbf;
hdl = get(MyEdgeFig, 'UserData');
v = hdl.Threshold;
s = get(hdl.ThreshCtrl,'String');
vv = real(evalin('base',['[' s ']'],num2str(v)));
if isempty(vv) | ~isreal(vv) | vv(1)<0
  vv = v;
  set(gcbo,'String',num2str(vv));
  return
end
vv = round(vv(1) * 1000000)/1000000;
set(gcbo,'String',num2str(vv));
hdl.Threshold = vv;
set(hdl.Apply, 'Enable', 'on');
setstatus(MyEdgeFig, '单击 ''运行程序按钮''计算图像边缘.');
set(MyEdgeFig, 'UserData', hdl);
return
% 函数 UpdateDirectionality
function UpdateDirectionality()
MyEdgeFig = gcbf;
hdl = get(MyEdgeFig, 'UserData');
v = get(hdl.sprDirPop,{'value','String'});
dir = deblank(v{2}(v{1},:));
set(hdl.sprDirPop, 'userdata', dir);
hdl.Directionality = dir;
set(hdl.Apply, 'Enable', 'on');
setstatus(MyEdgeFig, '单击 ''运行程序按钮''计算图像边缘.');
set(MyEdgeFig, 'UserData', hdl);
return
% 函数 Radio
function Radio(control, MyEdgeFig)
if nargin<2
  MyEdgeFig = gcbf;
end
hdl = get(MyEdgeFig, 'UserData');
if strcmp(control, 'auto')
  set(hdl.RadioAutomatic, 'Value', 1);
  set(hdl.RadioManual, 'Value', 0);
  set(hdl.ThreshCtrl, 'Enable', 'off');
  set(hdl.Apply, 'Enable', 'on');
  setstatus(MyEdgeFig, '单击 ''运行程序按钮''计算图像边缘.');
elseif strcmp(control, 'manual')
  set(hdl.RadioAutomatic, 'Value', 0);
  set(hdl.RadioManual, 'Value', 1);
  set(hdl.ThreshCtrl, 'Enable', 'on');
  set(hdl.Apply, 'Enable', 'on');
```

```
  setstatus(MyEdgeFig, '单击 ''运行程序按钮'' 计算图像边缘.');
end
return

% 函数 UpdateLOGSigma
function UpdateLOGSigma()
MyEdgeFig = gcbf;
hdl = get(MyEdgeFig, 'UserData');
v = hdl.LogSigma;
s = get(hdl.logSigmaCtrl,'String');
vv = real(evalin('base',s,num2str(v)));
if isempty(vv) | ~isreal(vv) | vv(1)<0
  vv = v;
  set(hdl.logSigmaCtrl,'String',num2str(vv));
  return
end
vv = round(vv(1) * 100)/100;
set(hdl.logSigmaCtrl,'String',num2str(vv));
hdl.LogSigma = vv;
set(hdl.Apply, 'Enable', 'on');
setstatus(MyEdgeFig, '单击 ''运行程序按钮'' 计算图像边缘.');
set(MyEdgeFig, 'UserData', hdl);
return

% 函数 ActivateSPRControls
function ActivateSPRControls(MyEdgeFig)
hdl = get(MyEdgeFig, 'UserData');
set([hdl.sprDirPop hdl.sprDirLbl], 'Visible', 'on');
set([hdl.logSigmaCtrl hdl.logSigmaLbl], 'Visible', 'off');
% 函数 ActivateLOGControls
function ActivateLOGControls(MyEdgeFig)
hdl = get(MyEdgeFig, 'UserData');
set([hdl.logSigmaCtrl hdl.logSigmaLbl],'Visible', 'on');
set([hdl.sprDirPop hdl.sprDirLbl], 'Visible', 'off');
```

11.8.2 应用举例

进入 MATLAB(7.0 及以上版本),选择 Debug 菜单中的 Run 命令运行 EdgeDection 程序,弹出如图 11-30 所示的 MATLAB Editor 对话框,选择 Add directory to the bottom of the MATLAB path 单选按钮即可运行程序。

程序 EdgeDection 可进行基于差分运算的二维图像边缘检测运算,程序运行界面如图 11-31 所示。

边缘检测程序 EdgeDection 实现了常用的边缘检测算子并可选择算子方向,各功能均以下拉菜单方式实现。应用该程序可方便地对原始图像、边缘检测算子及差分算子的方向等进行选择。例如,若原始图像选择土星图像,边缘检测方法选择 Sobel 算子,则原始图像及边缘检测的结果图像如图 11-32 所示。

MATLAB Editor

File C:\Documents and Settings\luan\My Documents\cth\图像处理文件\EdgeDetection.m is not found in the current directory or on the MATLAB path.

To run this file, select one of the following

- Change MATLAB current directory
- Add directory to the top of the MATLAB path
- Add directory to the bottom of the MATLAB path

OK　Cancel

图 11-30　MATLAB Editor 对话框

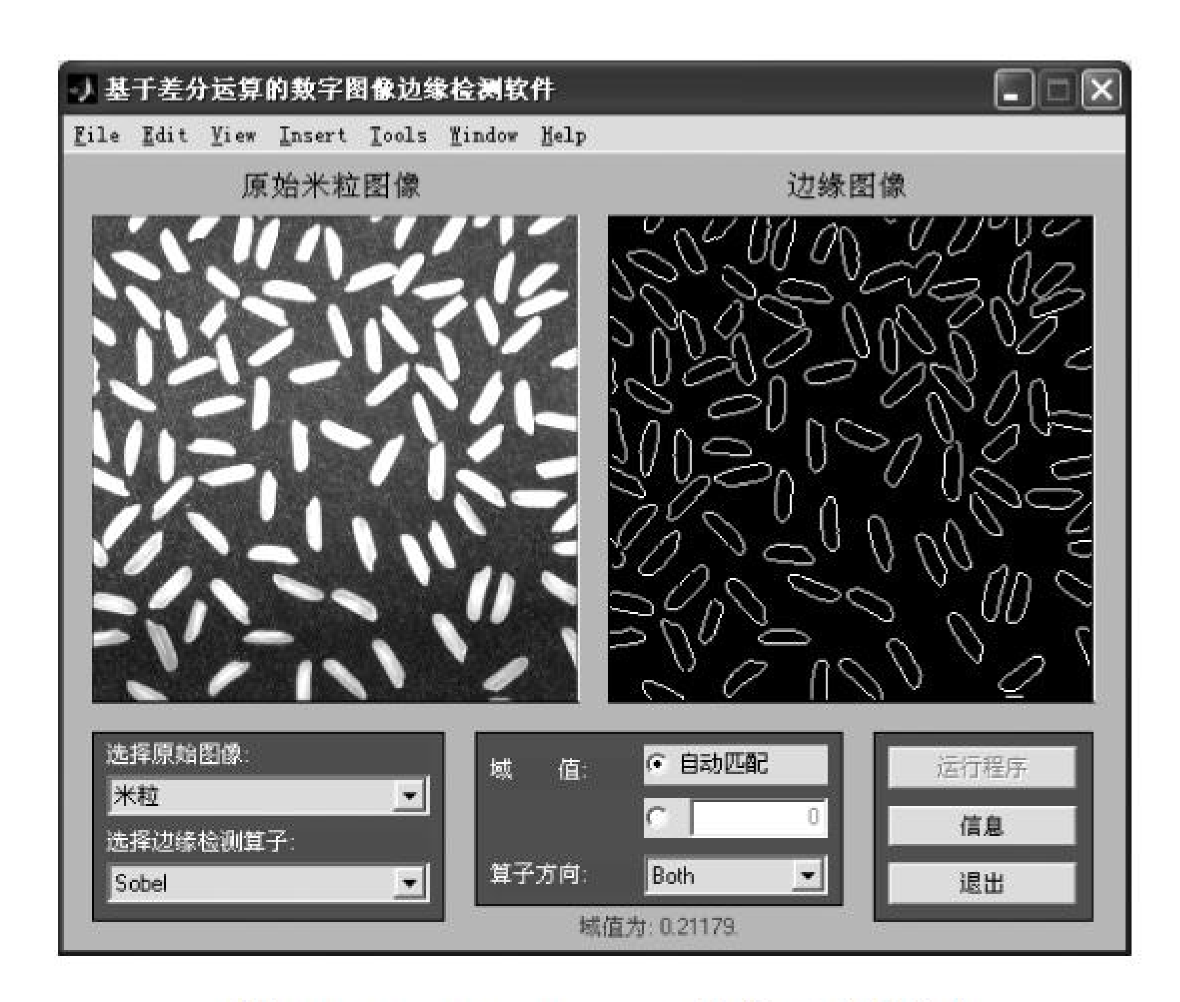


图 11-31　EdgeDection 程序运行结果



图 11-32　土星图像的运行结果

习题

1. 什么是图像分割？常用的图像分割方法可以分为哪几种类型？

2. 试简述灰度阈值分割算法的基本原理。

3. 根据灰度阈值分割法的基本原理，设计一个 MATLAB 程序找出具有双峰直方图特性图像的最佳分割阈值。

4. 若有一幅图像，已知其背景和目标灰度均服从正态分布，其均值分别为 μ 和 v，且图像与背景面积相等，试确定其最佳阈值。

5. 基于区域的图像分割方法主要包括哪几种算法？

6. 阐述区域分裂与合并法的实现过程。

7. 什么是图像的边缘？常见的边缘信号有哪几种类型？

8. 基于一阶差分运算的边缘检测算子主要包括哪些？各有何特点？

9. 基于二阶差分运算的边缘检测算子主要包括哪些？各有何特点？

10. 在数字图像边缘检测中，Laplace 算子有何特点？

11. Canny 提出的最优边缘准则是什么？

12. 试简述 Canny 边缘检测算法的步骤。

13. 选择一幅图像，编写 MATLAB 程序进行 Canny 边缘检测。

14. LoG 算子的基本原理是什么？它具有哪些特点和作用？

15. 应用本章的边缘检测程序 EdgeDetection 进行 3～6 种图像的边缘检测实验，分析各种算子的特点和适应性。

16. 在数字图像处理中，如何实现轮廓跟踪？

17. 根据模板匹配法中匹配误差测度的定义，分析其意义和作用。

18. 数字图像处理技术中常用的直方图匹配主要有哪几种方法？各有何特点？

第 12 章 图像特征与理解

CHAPTER 12

图像分割将一幅图像分解成若干个特定的、具有独特性质的区域，而图像特征与理解是指对分割以后的图像，提取图像内的物体或区域的特征，以用于图像识别、分类和理解等。图像特征是区分一个图像内物体或区域的最基本属性和特点。图像特征既可以是人的视觉能够识别的自然特征，也可以是通过对图像进行测量和处理，人为获取或定义的某些特征。这些特征可以包括几何特征和形状特征等，如物体的周长、面积，以及物体或区域的连通性、尺寸、形状等几何特性。物体的几何特性对描述图像具有非常重要的作用，图像的描述总是以被描述物体的某些特征为基础的。

图像分割完成之后，可以对图像内的物体进行几何特征测量和分析，进而用于对物体进行识别，或对物体进行分类。因此，图像特征与理解具有广泛的应用，例如，可以用于工农业生产中的质量监控，即根据识别的结果判别物体是否符合指定的标准。此外，对马铃薯或苹果等农产品按品质自动分类的机器视觉系统；将传送带上不同工件自动分类的视觉系统等都是基于图像特征与理解方面的应用，图像特征与理解也是机器视觉的主要内容之一。

本章围绕图像的特征与理解，首先介绍图像的几何特征、形状特征和彩色图像的颜色特征，然后介绍图像形状描述子、纹理特征描述以及曲线曲面拟合和图像的骨架提取等。

12.1 几何特征

图像的几何特征是指图像中物体的位置、方向、周长和面积等方面的特征。尽管几何特征比较直观和简单，但在图像分析与理解中可以发挥重要的作用。提取图像几何特征之前，一般要对图像进行分割和二值化处理。二值图像只有 0 和 1 两个灰度级，便于获取、分析和处理，虽然二值图像只能给出物体的轮廓信息，但在图像分析和计算机视觉中，二值图像及其几何特征特别具有价值，可用来完成分类、检验、定位、轨迹跟踪等任务。

12.1.1 位置与方向

1. 位置

一般情况下，图像中的物体通常并不是一个点，因此，采用物体或区域的面积的中心点作为物体的位置。如图 12-1 所示，面积中心就是单位面积质量恒定的相同形状图形的质心 O。

由于二值图像质量分布是均匀的，故质心和形心重合。若图像中的物体对应的像素位置坐标为$(x_i,y_i)(i=0,1,\cdots,N-1;\ j=0,1,\cdots,M-1)$，则可用下式计算质心位置坐标：

$$\begin{cases}\bar{x}=\dfrac{1}{NM}\sum\limits_{i=0}^{N-1}\sum\limits_{j=0}^{M-1}x_i\\ \bar{y}=\dfrac{1}{NM}\sum\limits_{i=0}^{N-1}\sum\limits_{j=0}^{M-1}y_i\end{cases}\tag{12-1}$$

2. 方向

图像分析不仅需要知道一幅图像中物体的具体位置，而且还要知道物体在图像中的方向。如果物体是细长的，则可以将较长方向的轴定义物体的方向。如图12-2所示，通常，将最小二阶矩轴定义为较长物体的方向。

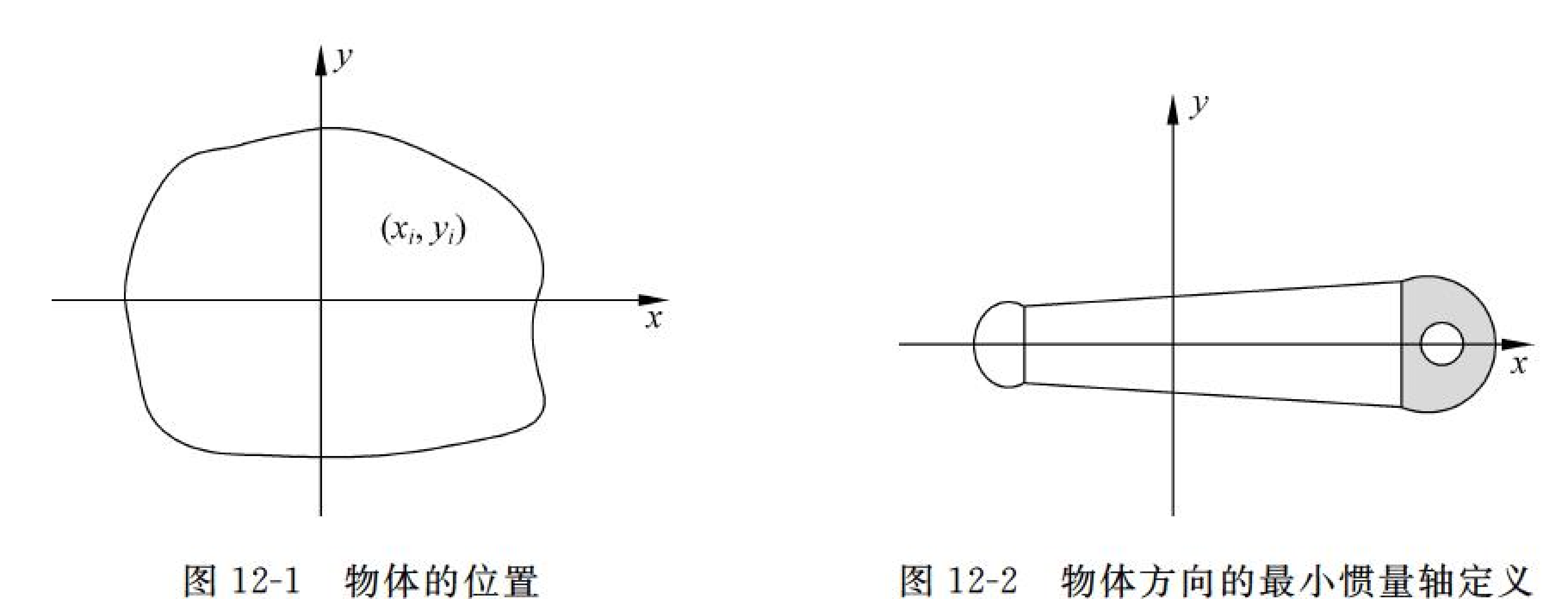


图12-1　物体的位置　　　图12-2　物体方向的最小惯量轴定义

也就是说，要找出一条直线，使下式的值最小：

$$E=\iint r^2 f(x,y)\mathrm{d}x\mathrm{d}y\tag{12-2}$$

式中，r是点(x,y)到直线(轴线)的垂直距离。通常情况下，确定一个物体的方向并不是一件容易的事情，需要进行一定量的测量。

12.1.2 长轴与短轴

若区域或物体的边界已知，则可以采用区域的最小外接矩形(mini-mum enclosing rectangle，MER)的尺寸来描述该区域的基本形状，如图12-3所示，a为长轴，b为短轴。显然，这是一种较简单的形状描述方法。当物体所占区域的方向与坐标轴方向一致时，只需计算物体边界点的最大和最小坐标值，就可得到物体的最小外接矩形、长轴和短轴。但对任意朝向的物体，长轴和短轴可能与坐标轴方向不一致，这时可使区域的边界旋转，根据精度要求，每旋转一定角度(如3°或5°等)，计算一次坐标轴方向上外接矩形(不一定是MER)的尺寸，即边界点最大最小x、y值距离，旋转到某一个角度后，外接矩形的面积达到最小，该最小面积的外接矩形就是MER，从而得出长轴和短轴数据。另外，还可以通过矩或求物体的最佳拟合直线的方法求出长轴和短轴数据。

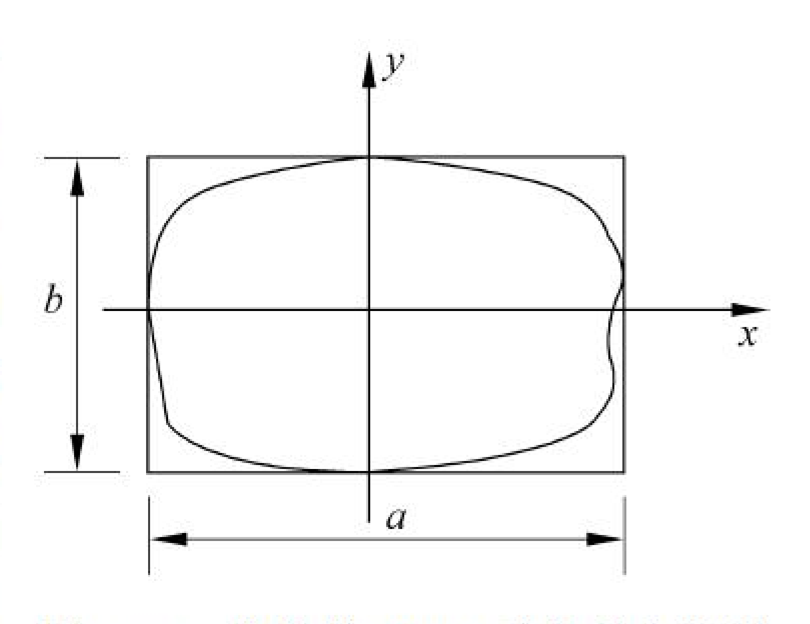


图12-3　物体的MER及长轴和短轴

12.1.3 周长

图像内某一物体或区域的周长是指该物体或区域的边界长度。一个形状简单的物体用相对较短的周长来包围它所占有面积内的像素，即周长是围绕所有这些像素的外边界的长度。通常，测量周长会包含物体内多个 90°的转弯，这些拐弯一定程度上扩大了物体的周长。物体或区域的周长在区别某些简单或复杂形状的物体时具有重要价值。

由于周长的表示方法不同，因而计算周长的方法也有所不同。计算周长常用的 3 种方法分别如下：

(1) 若将图像中的像素视为单位面积小方块时，则图像中的区域和背景均由小方块组成。区域的周长即为区域和背景缝隙的长度之和，此时边界用隙码表示，计算出隙码的长度就是物体的周长。如图 12-4 所示图形，边界用隙码表示时，周长为 24。

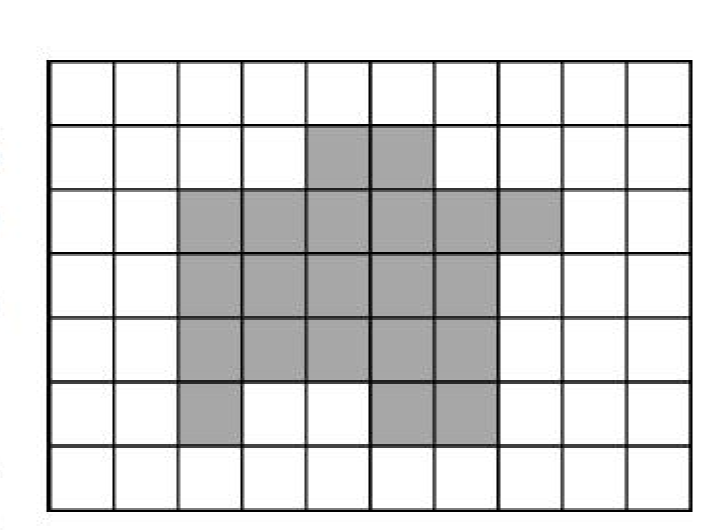


图 12-4 物体周长的计算

(2) 若将像素视为一个个点时，则周长用链码表示，求周长也就是计算链码的长度。当链码值为奇数时，其长度为$\sqrt{2}$；当链码值为偶数时，其长度为 1，即周长 p 可表示为

$$p = N_e + \sqrt{2} N_o \tag{12-3}$$

式中，N_e 和 N_o 分别是边界链码(8 方向)中走偶步与走奇步的数目。以图 12-4 为例，边界用链码表示时，物体的周长为 $10+5\sqrt{2}$。

周长也可以简单地从物体分块中通过计算边界上相邻像素的中心距离之和得出。

(3) 周长用边界所占面积表示时，周长即物体边界点数之和，其中每个点位占面积为 1 的一个小方块。以图 12-4 为例，边界以面积表示时，物体的周长为 15。

12.1.4 面积

面积是衡量物体所占范围的一种方便的客观度量。面积与其内部灰度级的变化无关，而完全由物体或区域的边界决定。同样面积条件下，一个形状简单的物体其周长相对较短。

1. 像素计数法

最简单的面积计算方法是统计边界及其内部的像素的总数。根据面积的像素计数法的定义方式，物体面积的计算非常简单，求出物体边界内像素点的总和即为面积。计算公式为

$$A = \sum_{x=1}^{N} \sum_{y=1}^{M} f(x, y) \tag{12-4}$$

对二值图像而言，若 1 表示物体的像素，而 0 表示背景像素，则面积就是统计 $f(x,y)=1$ 的像素数量。

2. 边界行程码计算法

由各种封闭边界区域的描述来计算面积也很方便，面积的边界行程码计算法可分如下两种情况：

(1) 若已知区域的行程编码，则只需将值为 1 的行程长度相加，即为区域面积。

(2) 若已知某区域封闭边界的某种表示，则相应连通区域的面积为区域外边界包围的面积与内边界包围的面积(孔的面积)之差。

若采用边界链码表示面积，面积计算方法如下：

设屏幕左上角为坐标原点，区域起始点坐标为(x_0, y_0)，则第k段链码终端的纵坐标y为

$$y_k = y_0 + \sum_{i=1}^{k} \Delta y_i \tag{12-5}$$

式中

$$\Delta y_i = \begin{cases} -1 & \varepsilon_i = 1,2,3 \\ 0 & \varepsilon_i = 0,4 \\ 1 & \varepsilon_i = 5,6,7 \end{cases} \tag{12-6}$$

ε_i是第i个码元。设

$$\Delta x_i = \begin{cases} -1 & \varepsilon_i = 3,4,5 \\ 0 & \varepsilon_i = 2,6 \\ 1 & \varepsilon_i = 0,1,7 \end{cases} \tag{12-7}$$

$$a = \begin{cases} \dfrac{1}{2} & \varepsilon_i = 1,5 \\ 0 & \varepsilon_i = 0,2,4,6 \\ -\dfrac{1}{2} & \varepsilon_i = 3,7 \end{cases} \tag{12-8}$$

则相应边界所包围的面积为

$$A = \sum_{i=1}^{n} (y_{i-1}\Delta x_i + a) \tag{12-9}$$

应用式(12-9)面积公式计算的面积，即以链码表示边界时边界内所包含的单元方格数。

3. 边界坐标计算法

面积的边界坐标计算法采用高等数学中的格林公式进行计算，在x-y平面上，一条封闭曲线所包围的面积为

$$A = \frac{1}{2}\oint (x\mathrm{d}y - y\mathrm{d}x) \tag{12-10}$$

其中，积分沿着该闭合曲线进行。对于数字图像，可将上式离散化，因此可得

$$\begin{aligned} A &= \frac{1}{2}\sum_{i=1}^{N}[x_i(y_{i+1} - y_i) - y_i(x_{i+1} - x_i)] \\ &= \frac{1}{2}\sum_{i=1}^{N}(x_i y_{i+1} - y_i x_{i+1}) \end{aligned} \tag{12-11}$$

式中，N为边界点数。

12.1.5 距离

图像中两点$P_1(x_1, y_1)$和$P_2(x_2, y_2)$之间的距离是重要的几何性质之一，图像处理中的距离度量方法在2.4.3节中已介绍，常用的主要包括欧氏距离(又称为欧几里得距离)、城市距离(又称为市区距离)和棋盘距离3种方法。

1. 欧几里得距离

$$d_{\Omega}(P_1,P_2)=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \tag{12-12}$$

这是几何学中常用的距离测量方法,在数字图像处理中,由于欧几里得距离包含有平方和开方运算,因而较少被采用。

2. 市区距离

$$d_4(P_1,P_2)=|x_1-x_2|+|y_1-y_2| \tag{12-13}$$

$d_4(P_1,P_2)$(包括下面介绍的 $d_8(P_1,P_2)$)由于计算较为简便,且为正整数,因此常用来测量距离。

3. 棋盘距离

$$d_8(P_1,P_2)=\max(|x_1-x_2|,|y_1-y_2|) \tag{12-14}$$

如图 12-5(a)所示,设中心点 O 所在的位置为 $P_1(x_1,y_1)$,显然,以 $P_1(x_1,y_1)$为起点的市区距离形成以 P_1 为中心的菱形距离图形。图 12-5(a)为 $d_4(P_1,P_2)$距离小于 3 时的市区距离菱形图。可见,$d_4(P_1,P_2)$是从 P_1 到 P_2 最短的 4 路径的长度。同样,以 P_1 为起点的棋盘距离形成以 P_1 为中心的正方形。图 12-5(b)所示为 $d_8(P_1,P_2)$小于等于 3 时的棋盘距离正方形图形。同样由图可见 $d_8(P_1,P_2)$是从 P_1 到 P_2 最短的 8 路径的长度。

			3			
		3	2	3		
	3	2	1	2	3	
3	2	1	0	1	2	3
	3	2	1	2	3	
		3	2	3		
			3			

(a)

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

(b)

图 12-5 $d_4(P_1,P_2)$和 $d_8(P_1,P_2)$表示方法

12.2 形状特征

物体的形状特征主要包括矩形度、宽长比、圆形度、球状性、不变矩和偏心率等。物体从图像中分割出来以后,将形状特征与几何特征结合起来,在机器视觉系统中起着十分重要的作用,它可以作为区分不同物体的依据之一。

12.2.1 矩形度

物体的矩形度指物体的面积与其最小外接矩形的面积之比值。如图 12-6 所示,矩形度反映了一个物体对其外接矩形的充满程度。

矩形度的定义为

$$R=\frac{A_o}{A_{\mathrm{MER}}} \tag{12-15}$$

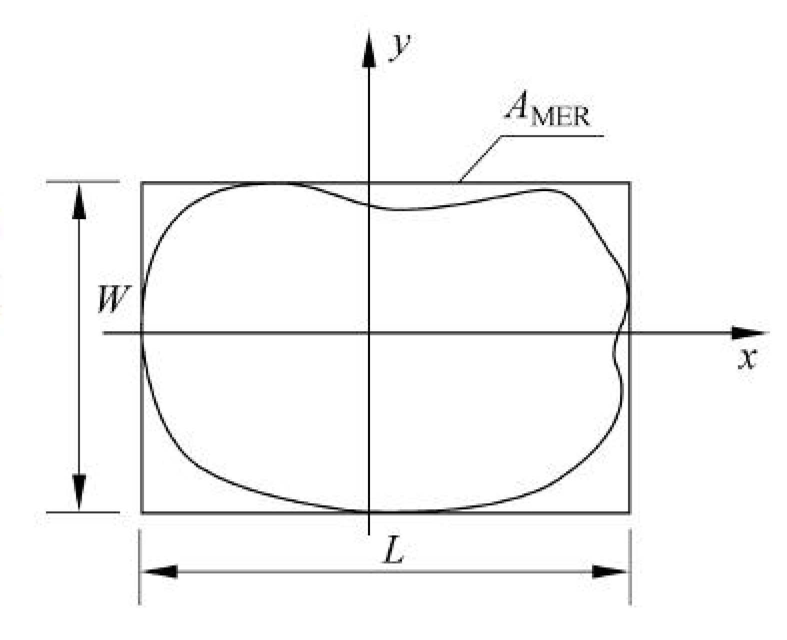


图 12-6 物体的最小外接矩形

式中,A_o 是该物体的面积；A_{MER} 是物体的最小外接矩

形(MER)的面积。

R 的值在 0~1 之间，当物体为矩形时，R 取得最大值 1.0。圆形物体的矩形度取值为 $\pi/4$，细长和弯曲物体的 R 值较小。

12.2.2 宽长比

宽长比是指物体的最小外接矩形的宽与长之比值。根据图 12-6 可得，宽长比 r 为

$$r = \frac{W}{L} \tag{12-16}$$

利用 r 可以将细长的物体与圆形或方形的物体区分开来。

12.2.3 圆形度

圆形度包括周长平方面积比、边界能量、圆形性、面积与平均距离平方之比值等。圆形度可以用来刻画物体边界的复杂程度。

1. 周长平方面积比

周长平方面积比是度量圆形度最常用的指标，又称为致密度，即周长的平方与面积之比值：

$$C = \frac{P^2}{A} \tag{12-17}$$

式中，P 表示物体的周长；A 表示面积。

2. 边界能量

边界能量是圆形度的重要指标之一。设物体的周长为 P，以变量 p 表示边界上的点到某一起始点的距离。边界上任意点都有一个对应的曲率半径，即该点与物体边界相切的圆的半径，如图 12-7 所示，以 $r(p)$ 表示曲率半径，则 p 点的曲率函数为

$$K(p) = \frac{1}{r(p)} \tag{12-18}$$

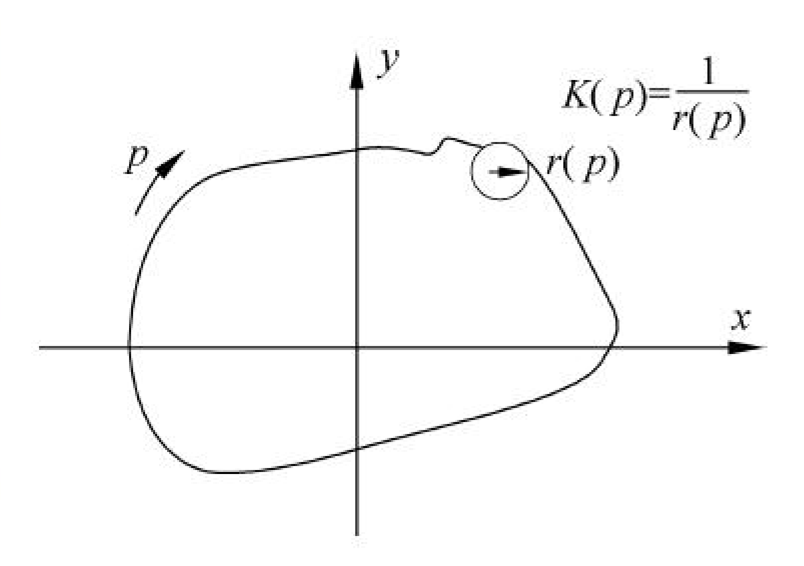


图 12-7 物体边界点的曲率半径

函数 $K(p)$ 是周期函数，其周期为 P。可采用下式计算单位边界长度的平均能量：

$$E = \frac{1}{P}\int_0^p |K(p)|^2 \mathrm{d}p \tag{12-19}$$

当不同物体具有相同面积时，圆具有最小边界能量：

$$E = \left(\frac{2\pi}{P}\right)^2 = \frac{1}{R^2} \tag{12-20}$$

式中，R 为圆的半径。曲率可以很容易地由链码算出，因而边界能量也可方便地求出。

3. 圆形性

圆形性(circularity)c 是一个用物体或区域 R 的所有边界点定义的特征量，即

$$c = \frac{\mu_R}{\delta_R} \tag{12-21}$$

式中

$$\mu_R = \frac{1}{K}\sum_{k=0}^{K-1} \|(x_k, y_k) - (\bar{x}, \bar{y})\| \tag{12-22}$$

μ_R 是从区域重心到边界点的平均距离；

$$\delta_R = \frac{1}{K}\sum_{k=0}^{K-1}[\|(x_k, y_k) - (\bar{x}, \bar{y})\| - \mu_R]^2 \tag{12-23}$$

δ_R 是从区域重心到边界距离的均方差。

当物体所占区域 R 趋向圆形时，特征量 c 是单调递增且趋向无穷，且不受区域平移、旋转和尺度变化的影响。

4. 面积与平均距离平方比值

面积与平均距离平方比值 g 也是圆形度的度量指标，其定义为

$$g = \frac{A}{\bar{d}^2} = \frac{N^3}{\sum_{i=1}^{N} x_i} \tag{12-24}$$

式中，x_i 是从具有 N 个点的物体中的第 i 个点到与其最近的边界点的距离。该定义利用了从边界上的点到物体内部某点的平均距离 $\bar{d}$，即

$$\bar{d} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{12-25}$$

12.2.4 球状度

球状度又称为球状性(sphericity)，它既可以用于描述二维物体也可以描述三维物体，球状度以 S 表示，其定义为

$$S = \frac{r_i}{r_c} \tag{12-26}$$

如图 12-8 所示，对于二维物体，r_i 代表区域内切圆(inscribed-circle)的半径，而 r_c 表示区域外接圆(circumscribed-circle)的半径，两圆的圆心都在物体区域的重心上。

当二维物体为圆形时，物体的球状性值 S 达到最大值 1.0，而当区域为其他形状时，则 $S<1.0$。球状性不受物体平移、旋转和尺度变化的影响。

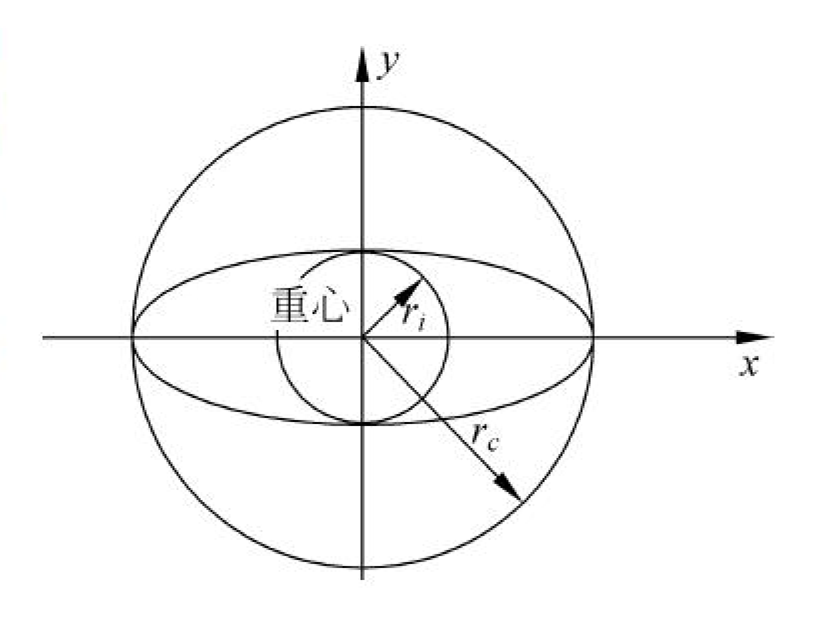


图 12-8 球状性定义示意图

12.2.5 不变矩

矩在物理学和工程上具有重要的理论意义和应用价值，图像像素的某些矩对于平移、旋转、尺度等几何变换具有许多不变特性，因此矩的表示方法在物体分类与识别中具有重要意义。

1. 矩的定义

对于二维图像函数 $f(x,y)$，其 $j+k$ 阶矩定义为

$$M_{jk} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^j y^k f(x,y)\mathrm{d}x\mathrm{d}y \quad j,k = 0,1,2,\cdots \tag{12-27}$$

j 和 k 可取所有的非负整数值，参数 $j+k$ 称为矩的阶。其中，零阶矩是物体的面积，即

$$M_{00} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\mathrm{d}x\mathrm{d}y \tag{12-28}$$

对二维离散函数序列 $f(x,y)$，零阶矩可表示为

$$M_{00}=\sum_{x=1}^{N}\sum_{y=1}^{M}f(x,y) \tag{12-29}$$

当 k 取不同值时，函数 $f(x,y)$ 的各阶矩形成了一个矩的无限集，即集合对于函数 $f(x,y)$ 是唯一的，根据该集合完全可以确定函数 $f(x,y)$。为了描述物体的形状，假设 $f(x,y)$ 的目标物体取值为 1，背景为 0，即函数只反映了物体的形状而忽略其内部的灰度级细节。

2. 质心坐标与中心矩

当 $j=1,k=0$，M_{10} 对二值图像而言即物体上所有点的横坐标 x 的总和，M_{01} 是物体上所有点的纵坐标 y 的总和。物体的各阶矩除以 M_{00} 以后，与物体的大小无关。而一阶矩除以 M_{00}，就是二值图像内一个物体的质心的坐标，即

$$\begin{cases}\bar{x}=\dfrac{M_{10}}{M_{00}}\\[2ex]\bar{y}=\dfrac{M_{01}}{M_{00}}\end{cases} \tag{12-30}$$

为了获得矩的不变特征，通常采用中心矩以及归一化的中心矩。中心矩的定义为

$$\mu_{jk}=\sum_{x=1}^{N}\sum_{y=1}^{M}(x-\bar{x})^{j}(y-\bar{y})^{k}f(x,y) \tag{12-31}$$

式中，$\bar{x}$、$\bar{y}$ 是物体的质心。如果 $\mu_{20}>\mu_{02}$，则区域可能是沿 x 轴方向伸长的区域，如 $\mu_{30}=0$，则区域关于 x 轴对称，如果 $\mu_{30}\geqslant 0$，则区域关于 y 轴对称，这种特征可用于测量简单的形状。由于中心矩以质心作为原点，具有位置无关性的特点。

3. 主轴

使二阶中心矩 μ_{11} 变得最小的旋转角 θ 可以由下式得出：

$$\tan 2\theta=\frac{2\mu_{11}}{\mu_{20}-\mu_{02}} \tag{12-32}$$

将 x、y 轴分别旋转角度 θ 得坐标轴 x'、y' 称为该物体的主轴。式(12-32)中，θ 为 90°时的不确定性可以通过如下限定条件解决：

$$\begin{cases}\mu_{20}<\mu_{02}\\[1ex]\mu_{30}>0\end{cases} \tag{12-33}$$

如果物体在计算矩之前旋转 θ 角，或相对于 x'、y' 轴计算矩，则矩具有旋转不变性。

4. 不变矩

相对于主轴计算采用面积归一化的中心矩，在物体放大、平移、旋转时保持不变，而三阶或更高阶的矩经过这样的规一化后不能保持不变性。

对于 $j+k=2,3,4,\cdots$，时称为高阶矩，对于高阶距可以定义如下归一化的中心矩：

$$\mu_{jk}=\frac{\mu'_{jk}}{(M_{00})^{r}} \tag{12-34}$$

式中

$$r=\left(\frac{j+k}{2}+1\right) \tag{12-35}$$

利用归一化的中心矩，可以获得不变矩的多种组合，这些组合对于平移、旋转、尺度等变换都具有不变性。

不变矩具备了一个好的形状特征所具有的某些性质。物体形状的唯一性体现在一个矩的无限集中，因此，要区别相似的形体需要一个很大的特征集。这样所产生的高维分类器对噪声和类内变化十分敏感。在某些情况下，其他阶数相对较低的矩可以反映一个物体的重要形状特征。不变矩特性已经应用于印刷体字符的识别、飞机形状区分、景物匹配和染色体分析之中。

12.2.6　偏心率

偏心率（eccentricity）又称为伸长度（elongation），它是区域形状的一种重要描述方法。偏心率在一定程度上反映了一个区域的紧凑性。偏心率有多种计算公式，一种常用的计算方法是区域长轴（主轴）长度与短轴（辅轴）长度的比值，如图 12-9 所示，即

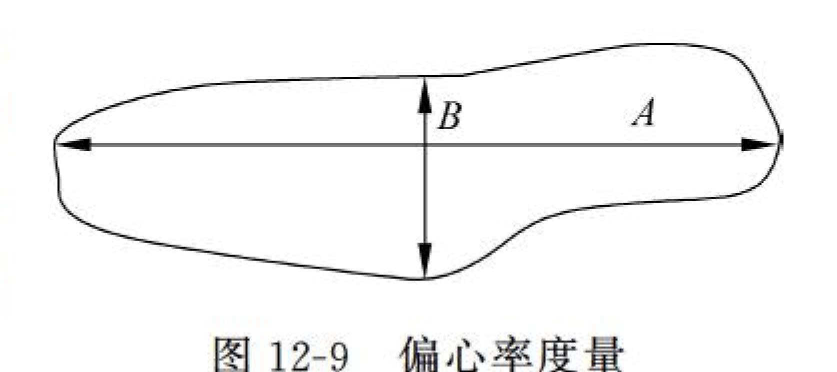


图 12-9　偏心率度量

$$E = \frac{A}{B} \tag{12-36}$$

图 12-9 中，长轴与短轴相互垂直，其长度分别是两个方向的最大值，这种定义受物体形状和噪声的影响较小。

偏心率的另一种方法是计算惯性主轴比，它基于边界线上的点或整个区域来计算质量。特南鲍姆（Tenebaum）提出了计算任意区域的偏心度的近似公式，一般过程如下：

1）计算平均向量

$$\begin{cases} \bar{x} = \dfrac{1}{N}\sum\limits_{i=1}^{N} x_i \\ \bar{y} = \dfrac{1}{N}\sum\limits_{i=1}^{N} y_i \end{cases} \tag{12-37}$$

2）计算 $j+k$ 阶中心矩

$$M_{jk} = \sum_{i=1}^{N}\sum_{i=1}^{N}(x_i - x_0)^j (y_i - y_0)^k \tag{12-38}$$

3）计算方向角

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2M_{11}}{M_{20} - M_{02}}\right) + \frac{\pi}{2}N \tag{12-39}$$

4）计算偏心度近似值

$$E = \frac{(M_{20} - M_{02}) + 4M_{11}}{A} \tag{12-40}$$

12.3　颜色特征

颜色特征是图像检索、图像识别与分类中应用广泛的重要视觉特征，主要原因在于颜色往往和图像中所包含的物体或场景十分相关。与其他视觉特征相比，颜色特征对图像自身的某些几何形状特性，如尺寸、方向、视角的依赖性较小，从而具有较高的鲁棒性。

颜色特征是一种全局特征，描述了图像或图像内区域所对应的景物的表面性质。颜色特征一般是基于像素点的特征，此时所有属于图像或图像区域的像素都有各自的贡献。由

于颜色对图像或图像区域的方向、大小等变化不敏感，所以颜色特征不能很好地捕捉图像中对象的局部特征。另外，仅使用颜色特征识别图像时，如果数据库很大，可能将许多不需要的图像也检索出来。颜色直方图是最常用的表达颜色特征的方法，其优点是不受图像旋转和平移变化的影响，基于归一化的直方图还可不受图像尺度变化的影响，其缺点是没有表达出颜色空间分布的信息。颜色特征常用的特征提取与匹配方法主要包括颜色直方图、颜色集、颜色矩、颜色聚合向量和颜色相关图。

12.3.1 颜色直方图

颜色直方图是很多图像检索系统中广泛采用的颜色特征，直方图描述了不同色彩在整幅图像中所占的比例，而并不关心每种色彩所处的空间位置，在几何特征上具有较好的鲁棒性。但颜色直方图由于失去了 R、G、B 亮度值的位置对应关系，因而无法描述图像中的对象或物体。因此，颜色直方图一般适合于描述一些难以进行自动分割的图像。

由于计算机具有多种颜色模型，因此，颜色直方图也可以基于不同的颜色空间和坐标系，最常用的颜色空间是 RGB 颜色空间，然而，RGB 空间结构并不符合人们对颜色相似性的主观判断。因此，基于 HSV 空间、Luv 空间和 Lab 空间的颜色直方图也常常被采用，这些颜色空间更接近于人们对颜色的主观认识。例如，在颜色直方图应用中，HSV 空间是直方图最常用的颜色空间，它的三个分量分别代表色彩(hue)、饱和度(saturation)和值(value)。

计算颜色直方图需要将颜色空间划分成若干个小的颜色区间(又称为一个柄)，每个小区间成为直方图的一个 bin。这个过程称为颜色量化(color quantization)，通过统计颜色落在每个小区间内的像素数量可以得到颜色直方图。颜色量化有很多方法，如向量量化、聚类方法或者神经网络方法都常被采用。常用的量化方法是将颜色空间的各分量进行均匀量化。聚类算法需要考虑图像颜色特征在整个空间中的分布情况，从而避免出现某些区间的像素数量非常稀疏的情况，使量化更为有效。另外，如果图像是 RGB 格式而直方图是基于 HSV 空间，则可以先建立从量化的 RGB 空间到量化的 HSV 空间之间的查找表(look-up table)，从而加快直方图的计算过程。

上述的颜色量化方法会产生一定的问题。设想两幅图像的颜色直方图几乎相同，只是互相错开了一个量化区间，这时如果采用 L_1 距离或者欧拉距离计算两者的相似度，会得到很小的相似度值。为了克服这个缺陷，需要考虑到相似但不相同的颜色之间的相似度。一种方法是采用二次式距离。另一种方法是对颜色直方图事先进行平滑过滤，即每个区间中的像素对于相邻的几个区间也有贡献。这样，相似但不相同颜色之间的相似度对直方图的相似度也有所贡献。

选择合适的颜色量化区间数目和颜色量化方法，应考虑具体应用的性能和效率要求。一般来说，颜色量化区间的数目越多，直方图对颜色的分辨能力就越强。然而，量化的区间数目过大，不仅颜色直方图的计算会增加很多，可能反而不利于在大型图像库中建立索引，对于某些特殊情况，使用过大的颜色量化方法不一定能够提高检索和识别效果。一种有效减少直方图颜色量化区间数的办法是只选用那些像素数量最多的子区间来构造图像特征，因为图像的主体颜色是由像素数量最大的若干子区间所呈现的。实验证明，这种方法并不会降低颜色直方图的检索效果，由于忽略了那些少数像素的区间，颜色直方图对噪声的鲁棒

性提高了,有时会使检索效果更好。

12.3.2 颜色集

颜色直方图法是一种基于纯统计学意义的全局颜色特征提取与匹配方法,较好地描述了图像在颜色方面的宏观特征信息,但却无法区分图像的局部颜色信息。为支持大规模图像库进行更精细的图像检索,Smith 和 Chang 用颜色集(color set)作为颜色直方图的一种近似和改进。

颜色集的提取方法是,首先将图像从 RGB 颜色空间转化成视觉均衡的颜色空间(如 HSV 空间),并将颜色空间量化成若干个柄。然后,用色彩自动分割技术将图像分为若干区域,每个区域用量化颜色空间的某个颜色分量来索引,从而将图像表达为一个二进制的颜色索引集。在图像匹配中,比较不同图像颜色集之间的距离和色彩区域的空间关系,包括区域的分离、包含和交。因为颜色集的表达为二进制的特征向量,便于构建二分查找树来加快检索速度,这对于大规模的图像数据库非常有利。

12.3.3 颜色矩

颜色是彩色图像最重要特征之一,被广泛用于图像检索中。但从图像中提取颜色特征时,很多算法都先要对图像进行量化处理。量化处理容易导致误检,尤其是图像特征维数越高,对检索越不利。

针对彩色图像检索的特征,Stricker 和 Oreng 提出了颜色矩的方法,颜色矩是一种简单有效的颜色特征表示方法,有一阶矩(mean)、二阶矩(viarance)和三阶矩(斜度,skewness)等,由于颜色信息主要分布于低阶矩中,所以用一阶矩、二阶矩和三阶矩足以表达图像的颜色分布,颜色矩已证明可有效地表示图像中的颜色分布。因此,图像的颜色矩一共只需要 9 个分量,即 R、G、B 3 个颜色分量,每个分量包括 3 个低阶矩。颜色矩的数学定义为

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \tag{12-41}$$

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-\mu_i)^2\right)^{\frac{1}{2}} \tag{12-42}$$

$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-\mu_i)^3\right)^{\frac{1}{3}} \tag{12-43}$$

式中,N 表示图像中像素点的总数;p_{ij} 表示第 i 个颜色通道中第 j 个灰度级的像素点出现的概率。

以 YUV 颜色空间为例,3 个颜色分量的 3 个低阶矩可以组成一个如下所示的 9 维直方图向量:

$$(\mu_Y, \sigma_Y, s_Y, \mu_U, \sigma_U, s_U, \mu_V, \sigma_V, s_V) \tag{12-44}$$

与其他的颜色特征相比,颜色矩具有简洁、方便的特点。该方法不需要颜色空间量化,特征向量维数低。但是,这种简单性也产生了缺点,实验发现该方法的检索效率比较低,实际应用中一般用颜色矩进行第一次检索,去掉那些在颜色上极不相似的图像,从而缩小检索范围。

12.3.4 颜色聚合向量

针对颜色直方图和颜色矩无法表达图像色彩的空间位置的缺点，Pass 提出了图像的颜色聚合向量(color coherence vector，CCV)作为图像特征。CCV 是颜色直方图的一种演变，其核心思想是将属于直方图每一个 bin 的像素分为两部分：如果该 bin 内的某些像素所占据的连续区域的面积大于给定的阈值，则该区域内的像素作为聚合像素，否则作为非聚合像素。假设 α_i 与 β_i 分别代表直方图的第 i 个 bin 中聚合像素和非聚合像素的数量，图像的颜色聚合向量可以表达为 $((\alpha_1,\beta_1),(\alpha_2,\beta_2),\cdots,(\alpha_N,\beta_N))$。而 $(\alpha_1+\beta_1,\alpha_2+\beta_2,\cdots,\alpha_N+\beta_N)$ 就是该图像的颜色直方图。

CCV 实现了将聚合像素和非聚合像素单独处理，从而保留了图像颜色的一定的空间分布信息，由于包含了颜色分布的空间信息，颜色聚合向量相比颜色直方图的检索效果更好，特别是对于具有大区域均匀分布的图像和大部分为纹理信息的图像，检索效果更好。

12.3.5 颜色相关图

颜色相关图(color correlogram)是图像颜色分布的另一种表达方式。这种特征不但刻画了某一种颜色的像素数量占整个图像的比例，还反映了不同颜色对之间的空间相关性。实验表明，颜色相关图比颜色直方图和颜色聚合向量具有更高的检索效率，特别是查询空间关系一致的图像。

如果考虑到任何颜色之间的相关性，颜色相关图会变得非常复杂和庞大，空间复杂度为 $O(N^2d)$。一种简化的变种是颜色自动相关图(color auto-correlogram)，它仅仅考察具有相同颜色的像素间的空间关系，因此空间复杂度降到 $O(Nd)$。

12.4 形状描述子

对物体进行描述时，有时希望能使用一些比单个参数提供更丰富的细节，而又比用图像本身更简明的方法来描述物体的形状，这就是形状描述子，它可以对物体形状进行简洁的描述。形状描述子包括傅里叶描述子、边界链码、微分链码等。

12.4.1 傅里叶描述子

物体边界的离散傅里叶变换序列可以作为定量描述边界形状的有效方法，傅里叶描述子(Fourier descriptor，FD)是目前形状表示方法中应用最多的描述子之一。采用傅里叶描述的优点是可以将二维问题转化为一维问题，即将 x-y 平面中的曲线段转化为一维函数，或将 x-y 平面中的曲线段转化为复平面上的一个序列，即将 x-y 平面与复平面重合，其中，实轴与 x 轴重合，虚轴与 y 轴重合。这样可用复数 $x_k+\mathrm{j}y_k$ 的形式来表示给定边界上的每个点 (x,y)。如图 12-10 所示，这两种表示实质是一致的，是点对点的一一对应映射关系。

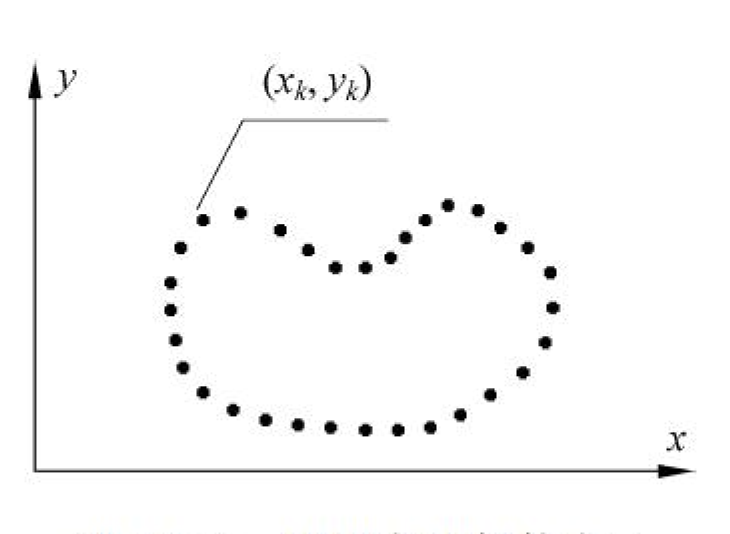


图 12-10　边界点的复数表示

根据以上分析，设物体的边界是由 N 个点组成的封闭边界，从任一点开始绕边界一周就得到一个复数序 $x_k+\mathrm{j}y_k$，即

$$s(k)=x_k+\mathrm{j}y_k \quad k=0,1,2,\cdots,N-1 \tag{12-45}$$

复数序列 $s(k)$ 称为傅里叶描述子(FD)。序列 $s(k)$ 的 DFT 变换为

$$S(\omega)=\frac{1}{N}\sum_{k=0}^{N-1}s(k)\mathrm{e}^{-\mathrm{j}2\pi\frac{k\omega}{N}} \quad \omega=0,1,2,\cdots,N-1 \tag{12-46}$$

傅里叶描述不必考虑其原始形状的大小、位置及方向。例如，若要改变轮廓大小，只要把 FD 乘一个常数就可以了。由于傅里叶变换是线性的，它的反变换也需乘以同样的常数。若将区域轮廓旋转一个角度，只需将每一个坐标乘以 $\mathrm{e}^{\mathrm{j}\theta}$ 就可以使其旋转 θ 角。由 DFT 的性质，在空域旋转了 θ 角，在频域中也会旋转 θ 角。关于轮廓起始点的移动，由 DFT 的周期性可以看到，在空域中的有限数字序列实际上代表周期函数的一个周期。

$S(\omega)$ 也可称为边界的傅里叶描述，其逆变换为

$$s(k)=\frac{1}{N}\sum_{\omega=0}^{N-1}S(\omega)\mathrm{e}^{-\mathrm{j}2\pi\frac{k\omega}{N}} \quad k=0,1,2,\cdots,N-1 \tag{12-47}$$

由于离散傅里叶变换是一种可逆线性变换，而且在变换过程中具有很多优秀的特性，因此，这一特点为边界描述提供了方便。若 $S(\omega)$ 只取频率域的 M 个值，即取 $S(\omega)$ 的前 M 个系数同样可求出 $s(k)$ 的一组近似值 $\hat{s}(k)$。

$$\hat{s}(k)=\frac{1}{N}\sum_{\omega=0}^{M-1}S(\omega)\mathrm{e}^{-\mathrm{j}2\pi\frac{k\omega}{N}} \quad k=0,1,2,\cdots,N-1 \tag{12-48}$$

式(12-48)中，虽然只取了频率域的 M 个值，时域边界 $s(k)$ 的 N 个值却可以全部求出，但 ω 的范围缩小了，即为重建边界点所用的频率项少了。由于傅里叶变换的高频分量对应物体的细节，而低频分量对应物体的基本轮廓。因此，采用适当数量的低频分量傅里叶系数就可以近似描绘出物体边界的近似形状。

在实际应用中还要考虑到如下一些问题：

(1) 如果采样不均匀将会给问题求解带来困难，因此，在理论上应采用等间隔取样。

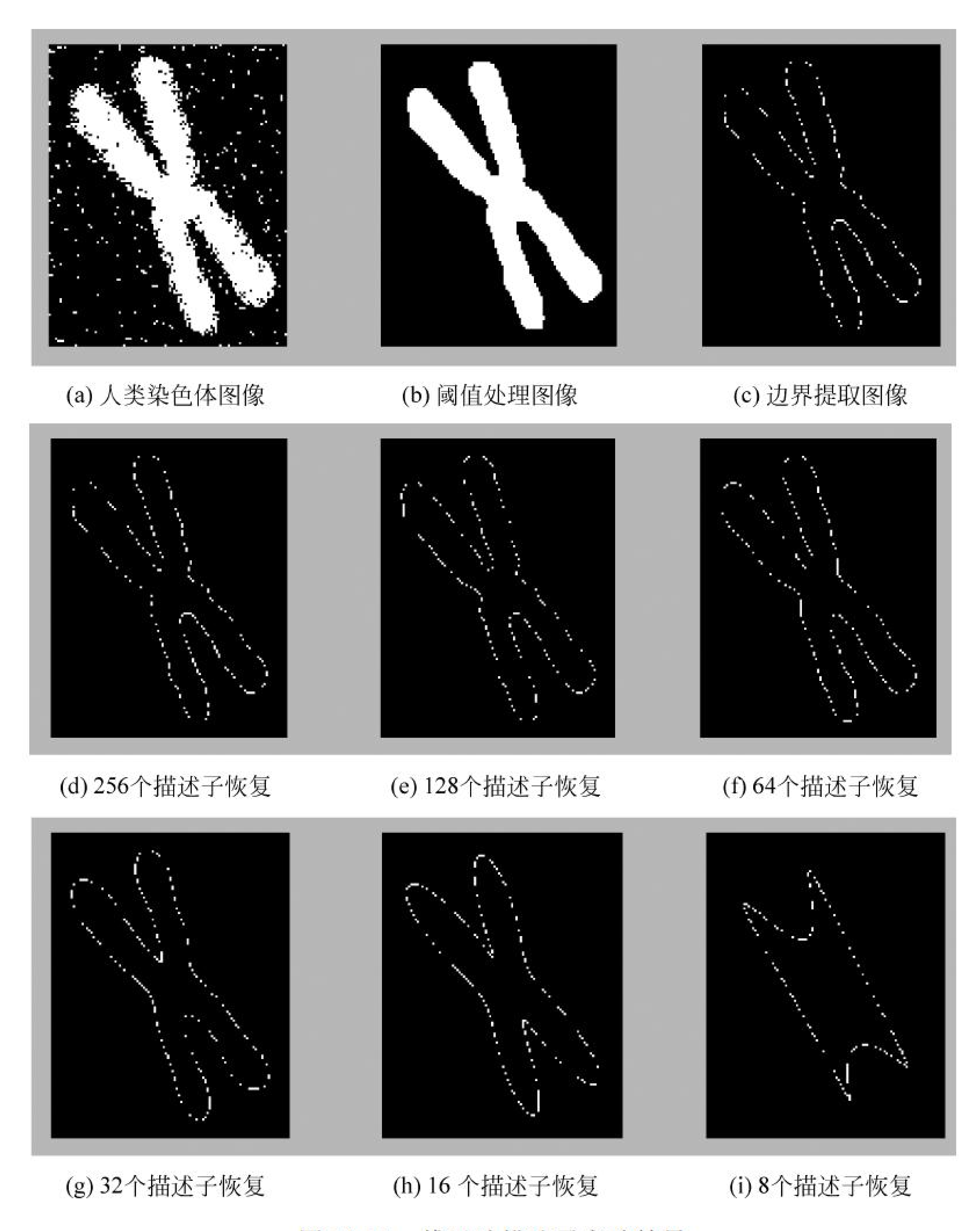
(2) FFT 的算法要求序列长度为 2 的整数次方，这样在采用 FFT 之前，应调整序列 $s(k)$ 的长度。如可先计算出轮廓的周长，则除以 2 的整数次方得出采样间隔，然后一个点、一个点进行追踪。

例 12-1：傅里叶描述子实验

由于傅里叶变换及其傅里叶描述子具有清晰的物理意义，本例应用傅里叶描述子对人类染色体图像进行傅里叶描述。图 12-11(a)为原始图像，图 12-11(b)～图 12-11(i)是采用不同数量的傅里叶系数对染色体图像进行描述的实验结果。本实验表明，仅用 16 个描述子，染色体外轮廓也较准确，当降至 8 个描述子时，对染色体外轮廓形状的描述发生了变化，出现了较大的误差。

12.4.2 拓扑描述

拓扑学(topology)是研究图形性质的理论。图形的拓扑性质具有稳定性，即只要图形没有发生破坏性变形，则其拓扑性质不会因为物理变形而改变。因此，区域的拓扑性质可用



(a) 人类染色体图像　(b) 阈值处理图像　(c) 边界提取图像

(d) 256个描述子恢复　(e) 128个描述子恢复　(f) 64个描述子恢复

(g) 32个描述子恢复　(h) 16 个描述子恢复　(i) 8个描述子恢复

图 12-11　傅里叶描述子实验结果

于对区域的全局描述,这些性质既不依赖于距离,也不依赖于距离测度的其他特性等。如图 12-12(a)所示,如果将区域中的孔洞数 H 作为拓扑描述子,显然,只要区域没有被撕裂或折叠,这个性质不受区域的伸长、旋转等方面的影响,孔洞数 H 就不会发生变化。

区域内的连接部分 C 的个数是区域的另一拓扑特性。一个集合的连通部分就是它的最大子集,在这个子集的任意点都可以用一条完全属于该子集中的曲线相连接。图 12-12(b)所示图形有三个连接部分。

欧拉数也是区域的重要拓扑特性之一,欧拉数(Euler-number)E 定义为

$$E = C - H \tag{12-49}$$

图 12-13(a)所示图像具有 1 个连接部分和 1 个孔,所以其欧拉数 E 为 0,而图 12-13(b)中,图像具有 1 个连接部分和 2 个孔,因而其欧拉数为−1。

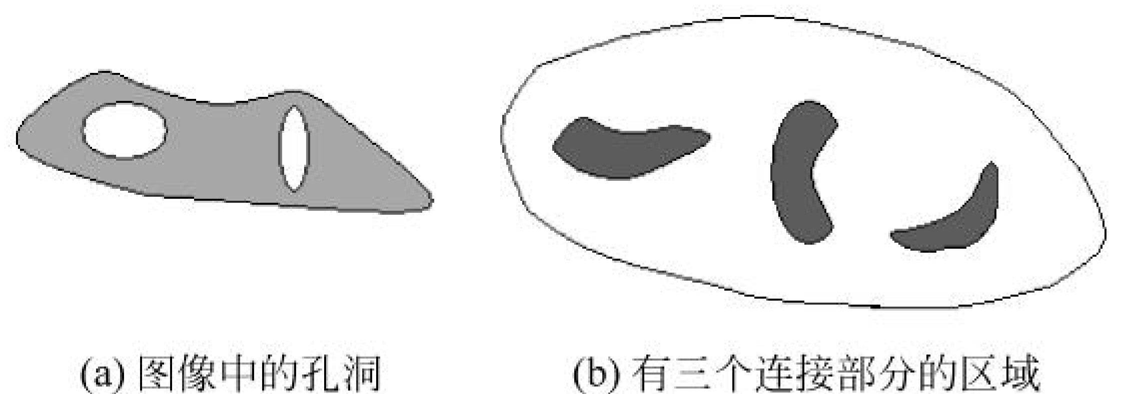
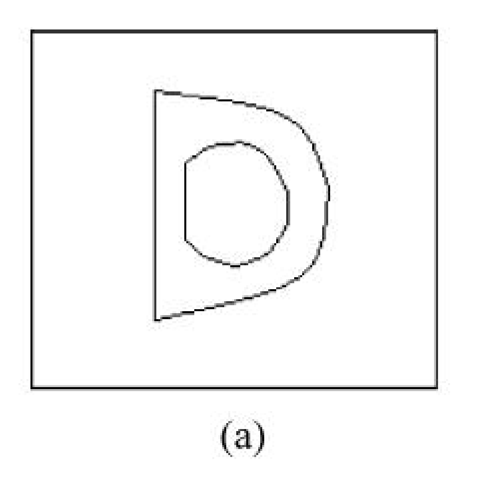


图 12-12　拓扑描述示意图



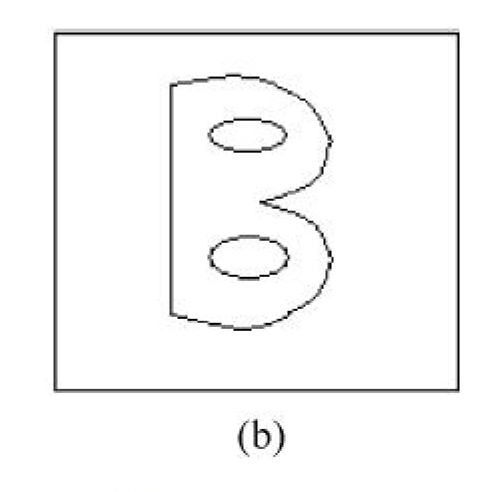


图 12-13　欧拉数的计算图形

12.4.3　边界链码

链码是对区域边界点的一种编码表示方法。该方法主要是利用一系列具有特定长度和方向的相连的直线段来表示目标的边界。由于每个线段的长度固定而方向数目有限，因此，仅仅有边界的起点需要采用绝对坐标表示，其余点都可只用接续方向来代表偏移量，而且对于每一个点只需一个方向数就可以代替两个坐标值，显而易见的是表示一个方向数比表示一个坐标值所需的比特数要少，因此采用链码表示可大大减少边界表示所需的数据量。

数字图像一般是按固定间距的网格进行采样的，因此最简单的链码是跟踪边界并赋给每两个相邻像素的连线一个方向值。常用的有 4 方向和 8 方向链码，如图 12-14(a)、(b)所示，其方向定义都为有限数量方向，而且直线段的长度是固定的。

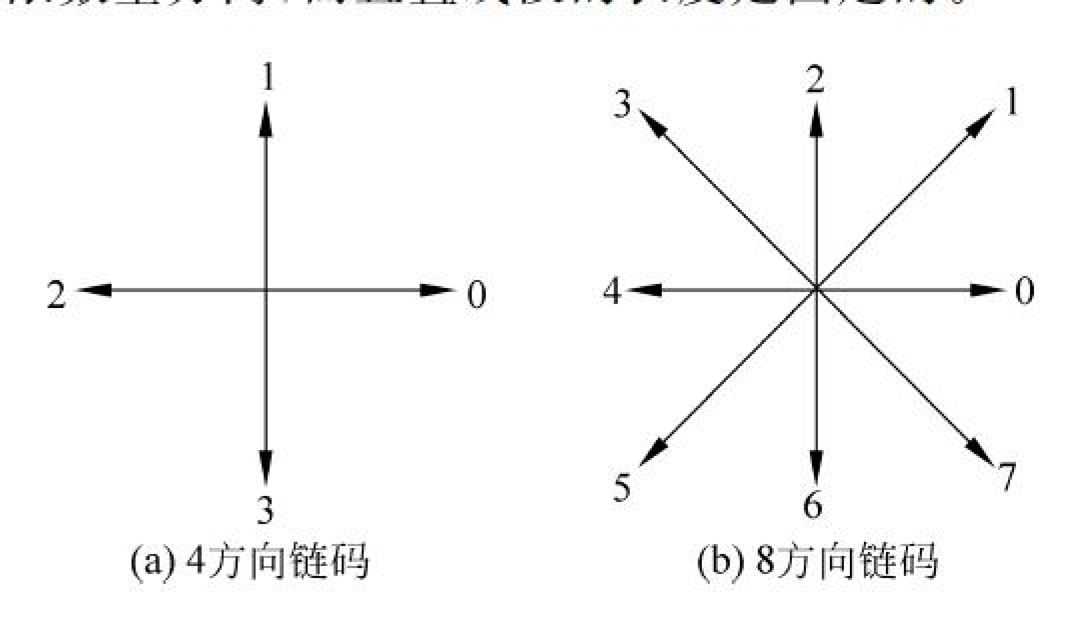


图 12-14　边界链码方向与码值

如图 12-15 所示的边界，若设起始点为 S，则可分别用 4 方向和 8 方向链码表示该区域的边界(逆时针方向)。

若采用 4 方向，则链码为 1　1　1　1　1　2　2　3　2　3　2　3　2　3　3　0　0　0　0　0。

若采用 8 方向，则链码为 2　2　2　2　2　4　4　6　4　6　4　6　4　6　6　0　0　0　0　0。

应用中直接对分割所得的目标边界进行编码可能会出现两个问题：一是码串比较长；

二是噪声及其他干扰可能导致边界小的变化，从而使链码发生与目标区域形状无关的较大变动。常用的改进方法是对原边界以较大的网格重新采样，并将与原边界点最接近的大网格点确定为新的边界点。这种方法也可用于消除目标尺度变化对链码的影响。

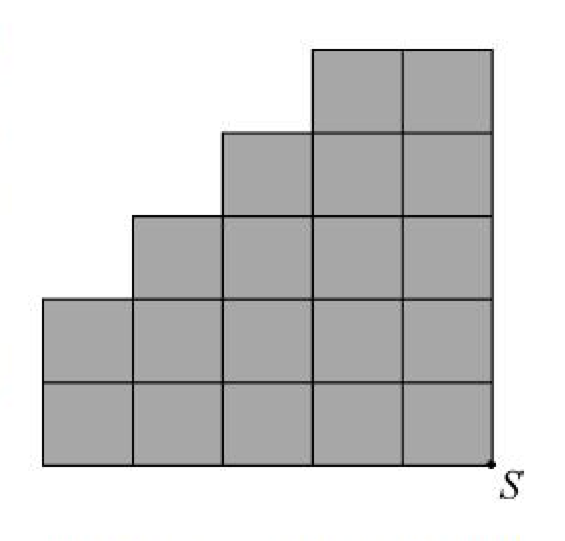


图 12-15 边界链码编码

使用链码时，起点的选择是很关键的。对同一个边界，如以不同的边界点作为链码的起点，得到的链码则是不同的。为解决这个问题可采用归一化链码表示方法，具体方法如下：

（1）给定一个从任意点开始产生的链码，先将它视为一个由各方向数组成的自然数。

（2）将这些方向数依一个方向循环，以使它们所构成的自然数的值最小。

（3）将这样转换后所对应的链码起点作为该区域边界的归一化链码的起点。

12.4.4 一阶差分链码

采用链码表示物体或区域边界的主要优点是当目标平移时，边界链码不会发生变化，而不足之处是，当区域旋转时则链码会发生变化。为解决旋转时链码变化的问题，可以采用链码旋转归一化处理方法，即应用原始链码的一阶差分来重新构造一个表示原链码各段之间方向变化的新序列。差分可用相邻两个方向数按反方向相减，所谓反方向即后一个减去前一个求取差分。如图 12-16(a)为原始图像的区域及方向数；图 12-16(b)为原始图像的链码及差分，其中上面一行为原链码，括号中的方向数为将最右一个方向数循环到左边，下面一行为上面一行的方向数两两相减得到的差分链码。

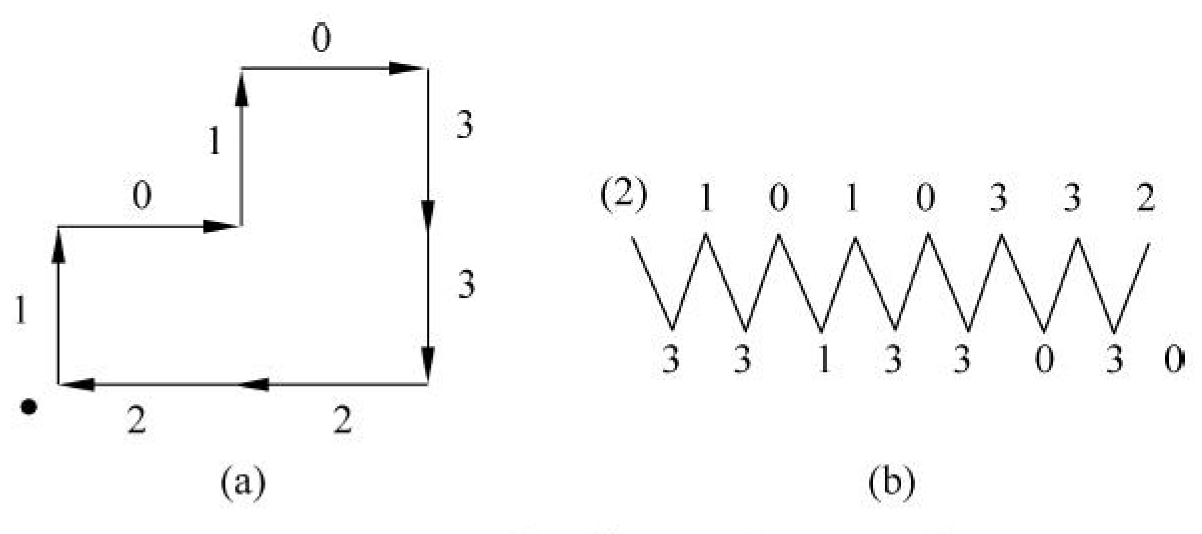


图 12-16 原始图像边界链码及差分

将图 12-16(a)逆时针旋转 90°得图 12-17，图 12-17 (a)、(b)分别为旋转以后图像的链码及差分。和图 12-16(b)一样，上面一行为 4 方向链码，括号中的方向数为将最右一个方向数循环到左边，下面一行为上面一行的方向数两两相减得到的差分链码，虽然原链码发生了变化，但归一化的差分码并没有变化。

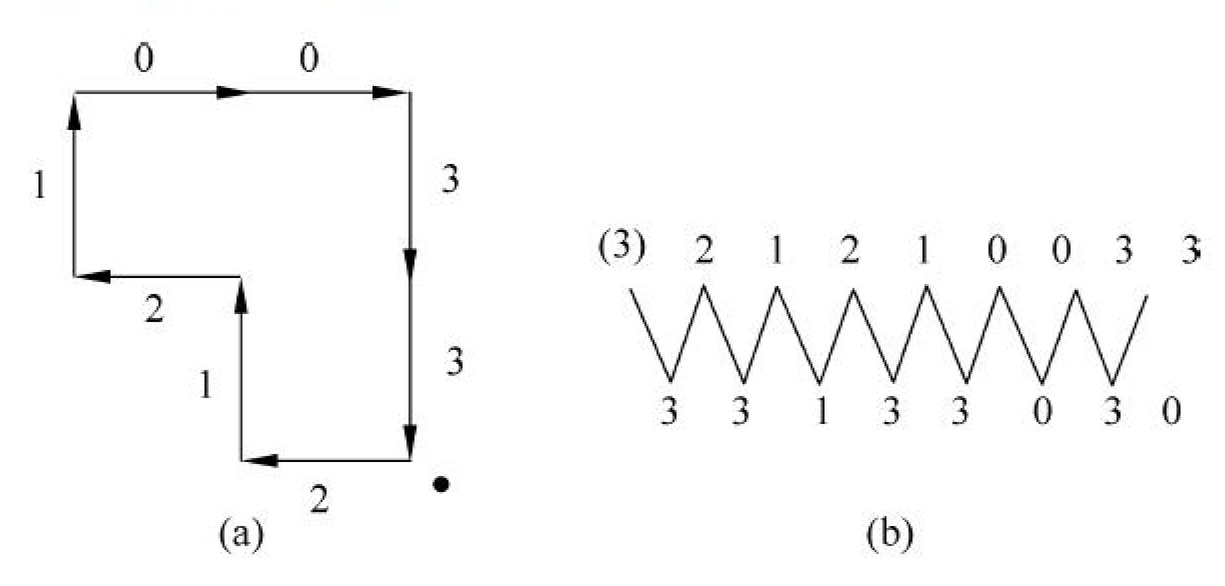


图 12-17 旋转图像边界链码及差分

12.4.5 霍夫变换

霍夫(Hough)变换是 Hough 于 1962 年提出的一种线描述方法，它通过笛卡儿坐标和极坐标两个坐标系之间的变换来检测平面内的直线和有规律的曲线。霍夫变换具有在变换空间所希望的边缘积聚而形成峰值点的现象。

霍夫变换可以将笛卡儿坐标空间的线变换为极坐标空间中的点。如图 12-18 所示，在 x-y 坐标系中的一条直线，若以 ρ 代表直线距原点的法线距离，θ 为该法线与 x 轴的夹角，则可用如下参数方程来表示该直线。

$$\rho = x\cos\theta + y\sin\theta \tag{12-50}$$

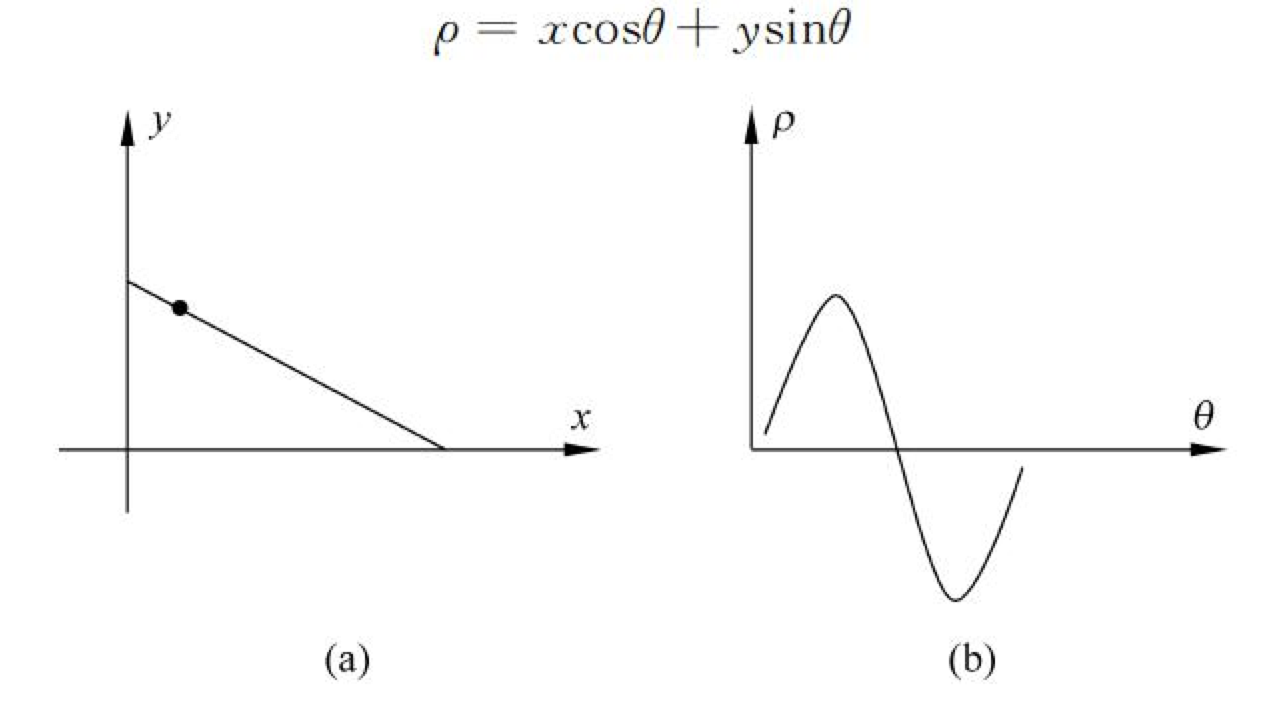


图 12-18　霍夫变换示意图

这就是直线的霍夫变换。通过霍夫变换将直角坐标系中的直线变换到极坐标系中则是一个点。而在 x-y 坐标系中的共点直线簇，映射到 (ρ,θ) 坐标系中就是一个点集。在 x-y 坐标系中，同一直线上的点映射到极坐标系就成为通过同一点的曲线簇。因此，霍夫变换使两个不同坐标系中的线和点建立了一种对应关系。

根据直角坐标系和极坐标系变换域之间的关系，总结霍夫变换具有如下主要性质：

(1) 直角坐标系中的一点对应于极坐标系中的一条正弦曲线。

(2) 变换域极坐标系中的一点对应于直角坐标系中的一条直线。

(3) 直角坐标系一条直线上的 N 个点对应于极坐标系中共点的 N 条曲线。这条性质可证明如下。

证明： 设 x-y 平面中在同一直线上的 N 个点分别为 (x_1,y_1)、(x_2,y_2)、…、(x_N,y_N)，因此，可得

$$y_i = ax_i + b \quad i = 1,2,3,\cdots,N \tag{12-51}$$

根据霍夫变换可知，对于上述 N 点，其变换域的曲线方程为

$$\rho_i = x_i\cos\theta + y_i\sin\theta \quad i = 1,2,3,\cdots,N \tag{12-52}$$

代入直线方程可得

$$\begin{aligned}\rho_i &= x_i\cos\theta_i + y_i\sin\theta_i, \quad i = 1,2,3,\cdots,N \\ &= x_i\cos\theta_i + (ax_i + b)\sin\theta_i \\ &= x_i(\cos\theta_i + a\sin\theta_i) + b\sin\theta_i\end{aligned} \tag{12-53}$$

由于 $x_i(i=1,2,3,\cdots,N)$ 可取直线上 N 个点中的任意一个，而且 N 本身可以取任意值，因此，极坐标系中的曲线一定通过 $\rho=b\sin\left[-\arctan\left(\frac{1}{a}\right)\right]$，$\theta=-\arctan\left(\frac{1}{a}\right)$ 这一点。

(4) 在极坐标系中，同一条曲线上的 N 个点对应于直角坐标系中通过同一点的 N 条直线。这条性质可证明如下。

证明：设极坐标系中有 N 个点，分别为

$$(\rho_1,\theta_1)、(\rho_2,\theta_2)、\cdots、(\rho_N,\theta_N)$$

由于 N 个点在同一曲线上，于是有

$$\rho_i = a\cos\theta_i + b\sin\theta_i \quad i = 1,2,3,\cdots,N \tag{12-54}$$

又因为

$$\rho_i = x_i\cos\theta_i + y_i\sin\theta_i$$

对于直角坐标系有

$$\begin{aligned} y_i &= -x_i\cot\theta_i + \frac{1}{\sin\theta_i}\rho \\ &= -x_i\cot\theta_i + \frac{1}{\sin\theta_i}(a\cos\theta_i + b\sin\theta_i) \\ &= (-x_i + a)\cot\theta_i + b \end{aligned} \tag{12-55}$$

因此，不论 θ_i 取值于极坐标系中的哪一个点，直线都经过 $x_i=a$，$y_i=b$ 这一点。

霍夫变换的应用可用如下方法实现：

在直角坐标中的每一离散数据点变换为极坐标系中的曲线。将 ρ 和 θ 分成许多小段，即将 (ρ,θ) 分成许多小单元，每一个小单元可设一累加器。在直角坐标系中可能落在直线上的每一点对应极坐标系中的一条曲线 $\rho_i=x_i\cos\theta_i+y_i\sin\theta_i$。分别使 θ 等于 $0,\Delta\theta,2\Delta\theta,\cdots$，便可求出相应的 ρ 值，并分别计算落在各小单元中的次数，当全部直角坐标点 (x_i,y_i) 完成变换后，可对小单元的累加器进行检测，峰值的小单元对应于极坐标曲线上的共线点，而这些曲线对应的直角坐标上的点可以认为是共线的。检测出直角坐标上的 N 点之后，将曲线交点坐标 (ρ_0,θ_0) 代入 $\rho_0=x\cos\theta_0+y\sin\theta_0$，就可以得到逼近 N 点的直线方程。

霍夫变换的优点是抗干扰能力强，当检测线上有小的扰动、断开或虚线时，经霍夫变换之后，在变换空间依然可以得到明显的峰值点。

12.5 纹理描述

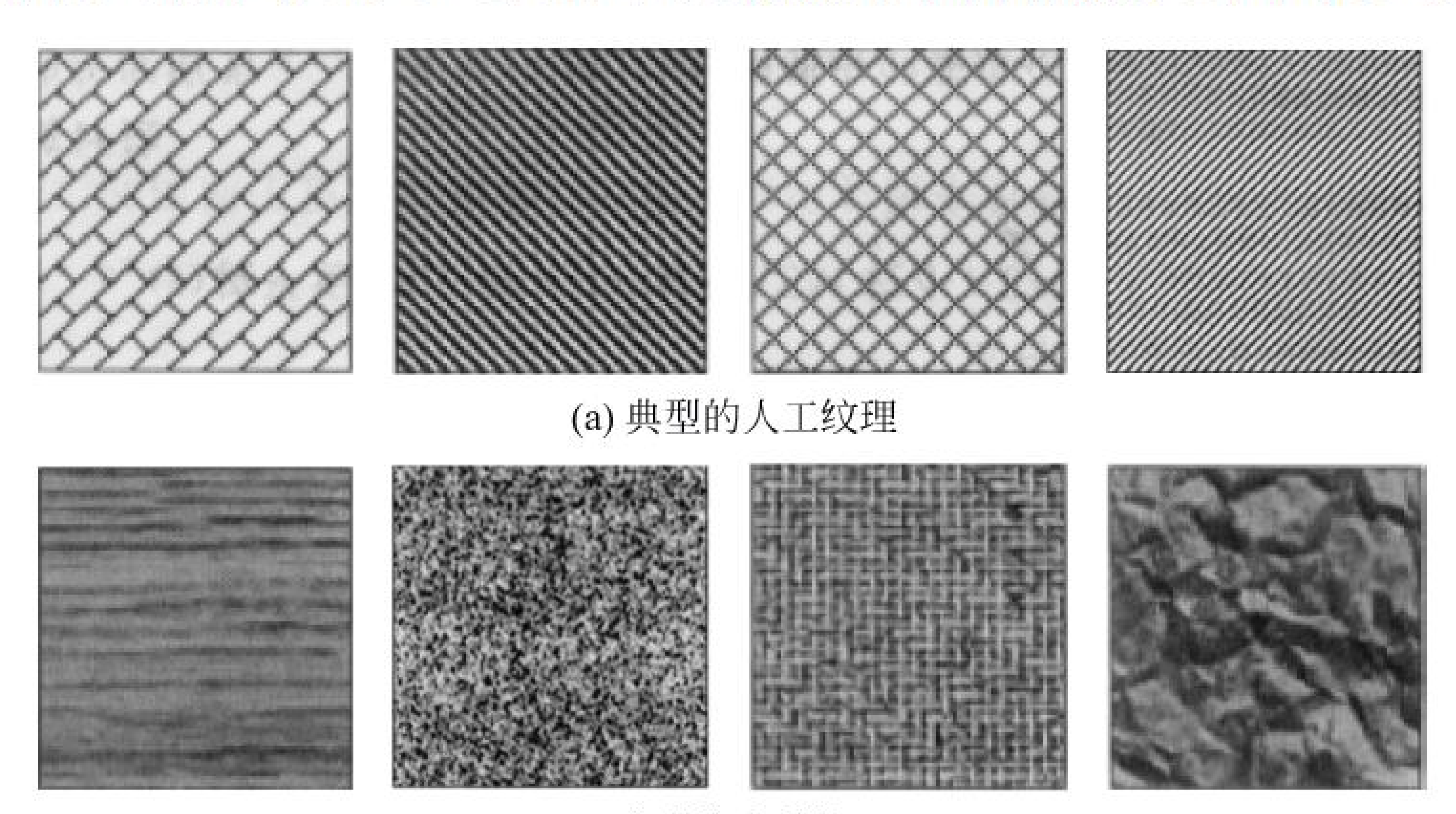
图像纹理是图像的重要特征，也是大家所熟悉的概念。什么是纹理，目前还没有统一、公认的确切定义。一般认为类似于布纹、草地、砖头、墙面等具有重复性结构的图像称为纹理图像。纹理图像在局部区域内可能呈现不规则性，但整体上则表现出一定的规律性，其灰度分布往往表现出某种周期性。纹理图像所表现出的这种特有的性质称为纹理。实际中很多图像具有纹理型结构，对这类纹理型图像可以通过纹理分析提取其宏观特征信息。

12.5.1 纹理特征

纹理(texture)一词最初指纤维物的外观，纹理图像在很大范围内没有重大细节变化，在这些区域内图像往往显示出重复性结构。有时，物体在纹理上与其周围背景和其他物体有区别，这时，图像分割应以纹理为基础。虽然纹理目前尚无统一的定义，但一般来说，纹理是由许多相互接近的、互相交织的元素构成，它们具有周期性。纹理在一定程度上反映了一

个区域中像素灰度级的空间分布的属性。

纹理可分为人工纹理和天然纹理(自然纹理)。人工纹理由某种符号的有序排列组成，这些符号可以是线条、点、字母、数字等。自然纹理是具有重复排列现象的自然景象，如砖墙、种子、森林、草地之类的照片。图 12-19(a)是常见的人工纹理结构，图 12-19(b)是常见的自然纹理结构。由此可见，人工纹理往往是有规则的，而自然纹理往往是无规则的。



(a) 典型的人工纹理

(b) 典型的自然纹理

图 12-19　典型的纹理结构图

归纳起来，对纹理的认识有两种看法。一种是凭人们的直观印象，而另一种是凭图像本身的结构。从直观印象出发的方法包含有心理学因素，这样就会产生多种不同的统计纹理特性，从这一观点出发，纹理分析应该采用统计方法。从图像结构观点出发，则认为纹理是一种结构，根据这一观点，纹理分析应该采用句法结构方法。

描述图像特性的参数有很多种，对于纹理图像来说有必要知道各个像素及其邻近像素的灰度分布情况。了解邻近像素灰度值变化情况的最简单方法是取 1 阶微分、2 阶微分的平均值与方差，如果要考虑纹理的方向性特征，则可考察 θ 方向与 $(\theta+\pi/2)$ 方向差分的平均值与方差。纹理分析常用的方法有统计法、自相关函数法、傅里叶频谱法、联合概率矩阵法和句法结构法等。

12.5.2　统计法

统计法是指利用图像内某一区域或物体的灰度直方图的矩对纹理结构进行描述，它又可以分为灰度差分统计法和行程长度统计法。

1. 灰度差分统计法

取图像内任意一点 (x,y)，设与该点相邻的点 $(x+\Delta x, y+\Delta y)$ 的灰度差值为

$$g_\Delta(x,y) = g(x,y) - g(x+\Delta x, y+\Delta y) \tag{12-56}$$

$g_\Delta(x,y)$ 称为灰度差分。若 $g_\Delta(x,y)$ 的所有可能取值共有 m 级，令点 (x,y) 在整个区域上移动，统计出 $g_\Delta(x,y)$ 取各个灰度级的次数，由此作出 $g_\Delta(x,y)$ 的直方图。根据直方图可以得出 $g_\Delta(x,y)$ 取不同灰度值的概率 $p_\Delta(i)$。

若 i 取值较小，而概率 $p_\Delta(i)$ 较大时，则说明纹理较粗糙；若概率 $p_\Delta(i)$ 较平坦，则说明纹理较细密。

灰度差分统计法一般采用以下参数描述纹理图像的特征：

1）对比度

$$\mathrm{CON} = \sum_i i^2 p_\Delta(i) \tag{12-57}$$

2）角度方向二阶矩

$$\mathrm{ASM} = \sum_i [p_\Delta(i)]^2 \tag{12-58}$$

3）熵

$$\mathrm{ENT} = -\sum_i p_\Delta(i) \lg p_\Delta(i) \tag{12-59}$$

4）平均值

$$\mathrm{MEAN} = \frac{\sum_i i p_\Delta(i)}{m} \tag{12-60}$$

根据上述公式，若 $p_\Delta(i)$较平坦，则 ASM 较小，而 ENT 较大；若 $p_\Delta(i)$分布在原点附近，则 MEAN 值较小。

2. 行程长度统计法

设图像内任意点(x,y)的灰度值为 g，与其相邻点的灰度值也可能为 g 或其他值，统计出从任一点出发沿 θ 方向上连续 n 个点都具有灰度值 g 所发生的概率，记此概率为 $P(g,n)$。在同一方向上具有相同灰度值的像素点的数量称为行程长度。根据 $P(g,n)$可以定义以下参数来描述纹理特征。

1）长行程加重法

$$\mathrm{LRE} = \frac{\sum_{g,n} n^2 p(g,n)}{\sum_{g,n} p(g,n)} \tag{12-61}$$

2）灰度值分布

$$\mathrm{GLD} = \frac{\sum_g \left[\sum_n p(g,n)\right]^2}{\sum_{g,n} p(g,n)} \tag{12-62}$$

3）行程长度分布

$$\mathrm{RLD} = \frac{\sum_g \left[\sum_n p(g,n)\right]}{\sum_{g,n} p(g,n)} \tag{12-63}$$

4）行程比

$$\mathrm{RPG} = \frac{\sum_{g,n} p(g,n)}{N^2} \tag{12-64}$$

式中，N^2 为图像的像素总数。

12.5.3 频谱法

频谱法即用傅里叶变换分析纹理的特征，依据傅里叶频谱的频率特性来描述周期的或近似周期的二维图像纹理结构。对于图像而言，如下二维傅里叶变换

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y \tag{12-65}$$

能包括其全部纹理信息。二维傅里叶变换的功率谱为

$$E=|F(u,v)|=FF^{*} \tag{12-66}$$

傅里叶频谱中突起的峰值对应纹理模式的主方向,峰值在频域平面的位置对应模式的基本周期,若采用滤波将周期性成分滤除,剩下的非周期性部分可用统计方法描述。

实际应用中,一般将频谱先转换到极坐标系中,如图 12-20 所示,此时傅里叶变换可用 $F(r,\theta)$表示,其频谱可用函数 $S(r,\theta)=|F(r,\theta)|^2$ 表示。

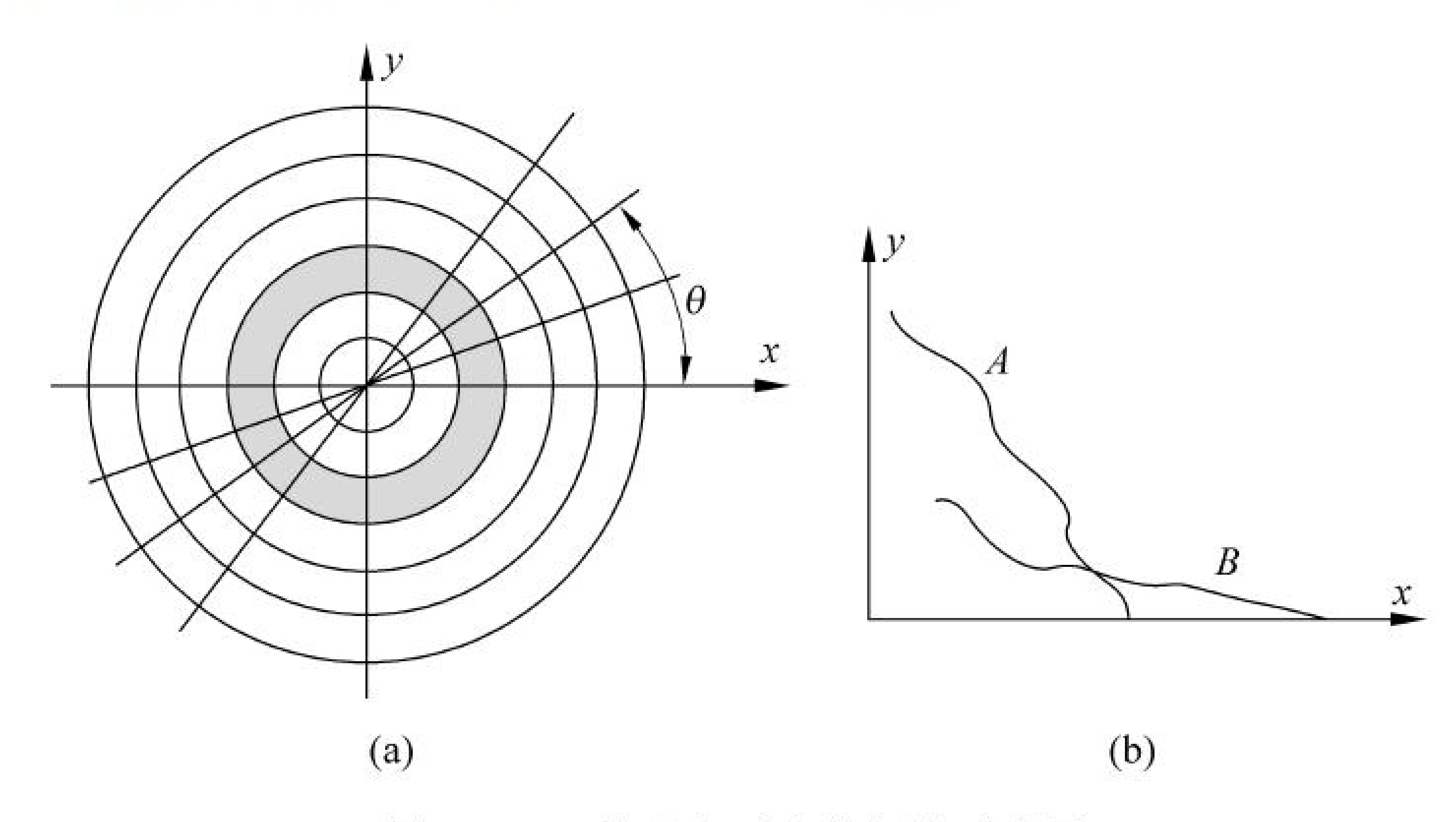


图 12-20 纹理和对应的频谱示意图

对每个确定的方向 θ,$S(r,\theta)$是一个一维函数 $S(r)$;对每个确定的频率 r,$S(r,\theta)$是一个一维函数 $S(\theta)$。对给定的 θ,分析 $S(r)$可得到频谱沿原点射出方向的行为特性;对给定的 r,分析 $S(\theta)$可得到频谱在以原点为中心的圆上的行为特性。如果将这些函数积分可得到更为全局性的描述,即

$$S(r)=\int_{0}^{2\pi}[F(r,\theta)]^2\mathrm{d}\theta \tag{12-67}$$

$$S(\theta)=\int_{0}^{\infty}[F(r,\theta)]^2\mathrm{d}r \tag{12-68}$$

如果是离散图像,将上述积分分别以求和代替即可得出相应的公式,即这些函数序列分别对 θ 或 r 求和同样可得到沿某一角度 θ 方向和半径 r 方向的全局性的描述,即

$$S(r)=\sum_{\theta}|F(r,\theta)|^2=\sum_{\theta}S(r,\theta) \tag{12-69}$$

$$S(\theta)=\sum_{r}|F(r,\theta)|^2=\sum_{r}S(r,\theta) \tag{12-70}$$

$S(r)$和 $S(\theta)$构成整个图像或图像区域的纹理频谱能量描述。如果 r 较小,$S(r)$很大,r 很大,$S(r)$却较小,则说明是粗糙性纹理结构;反之,如果 r 变化对 $S(r)$的影响较小,则为细密的纹理结构。在纹理粗糙的情况下,能量多集中在离原点很近的范围内,即图 12-20(b)中的曲线 A,而在纹理较细的情况下,能量分布在离原点较远的范围内,即图 12-20(b)中的曲线 B。

12.5.4 空间自相关函数法

物体的纹理常用其粗糙性加以描述。例如,在相同的外观条件下,毛织品一般比丝织品粗糙。粗糙性的程度与局部结构的空间重复周期有关,周期大的纹理细;反之则纹理粗糙。这种感觉上的粗糙虽不足以定量表示纹理的测度,但可说明纹理测度的变化趋势。即纹理测度值小表示纹理比较细密,纹理测度值大表示纹理比较粗糙。

设图像以 $f(m,n)$表示,则自相关函数可定义为

$$C(\varepsilon,\eta,j,k)=\frac{\sum_{m=j-w}^{j+w}\sum_{n=k-w}^{k+w}f(m,n)f(m-\varepsilon,n-\eta)}{\sum_{m=j-w}^{j+w}\sum_{n=k-w}^{k+w}[f(m,n)]^2} \tag{12-71}$$

式(12-71)是对$(2w+1)\times(2w+1)$窗口内的每一个像素点(j,k)与偏离值为 $\varepsilon,\eta=0,\pm1,\pm2,\cdots,\pm T$ 的像素之间的相关值进行计算。一般纹理区对给定偏离(ε,η)时的相关性要比细纹理区高,因而纹理粗糙性与自相关函数的扩展成正比。自相关函数扩展的一种测度就是二阶矩,定义为

$$T(j,k)=\sum_{\varepsilon=-T}^{j}\sum_{\eta=-T}^{k}\varepsilon^2\eta^2C(\varepsilon,\eta,j,k) \tag{12-72}$$

纹理越粗糙,则 $T(j,k)$越大,因此可以方便地用 $T(j,k)$作为度量纹理粗糙程度的参数之一。

12.5.5 灰度共生矩阵法

灰度共生矩阵法又称为联合概率矩阵法,是通过对图像的所有像素进行统计并描述其灰度分布的一种方法。取图像中任意一点(x,y)及偏离它的另一点$(x+a,y+b)$组成一个点对,设该点对的灰度值为(g_1,g_2)。令点(x,y)在所分析的区域内移动,则可得到全部(g_1,g_2)值。若灰度值的级数为 k,则(g_1,g_2)共有 k^2 种组合。对于整个区域,统计出每一个(g_1,g_2)值出现的次数,然后排列成一个方阵,再用(g_1,g_2)出现的总次数将之归一化为出现的概率 $p(g_1,g_2)$,这样的方阵称为联合概率矩阵,又称为灰度共生矩阵或灰度共现矩阵。

图 12-21 为一个简单的例子。图 12-21(a)为原始图像,共有 16 个灰度级,为使概率矩阵简单起见,首先将图 12-21(a)的灰度级数减为 4 级变为图 12-21(b)的形式。(g_1,g_2)分别取值为 0、1、2、3。因此将(g_1,g_2)各种组合出现的次数排列起来,可以得到图 12-21(c)~(e)所示的联合概率矩阵。以图 12-21(c)中的频数 10 为例,该数据表示图 12-21(b)中灰度值为(0,1)共出现 10 次,其余可依此类推。

由此可见,差分值(a,b)取不同的数值,就可以得到不同情况下的联合概率矩阵。(a,b)取值要根据纹理周期分布的特性进行选择,对于较细的纹理,选取(1,0)、(1,1)、(2,0)等较小的差分值,即 a、b 取值较小时,对应于变化缓慢的纹理结构,其联合概率矩阵对角线上的数值较大。若纹理的变化越快,则对角线上的数值就越小,对角线两侧的元素值却增大。为了能描述纹理的综合状况,应根据联合概率矩阵的特点计算纹理的相关特征参数,典型的参数有如下几种。

2	6	10	14	2	6	10
6	10	14	2	6	10	14
10	14	2	6	10	14	2
14	2	6	10	14	2	6
2	6	10	14	2	6	10
6	10	14	2	6	10	14
10	14	2	6	10	14	2

(a)

0	1	2	3	0	1	2
1	2	3	0	1	2	3
2	3	0	1	2	3	0
3	0	1	2	3	0	1
0	1	2	3	0	1	2
1	2	3	0	1	2	3
2	3	0	1	2	3	0

(b)

(g_1)				
0	0	10	0	0
1	0	0	11	0
2	0	0	0	11
3	10	0	0	0

→ (+1)
(a=1, b=0)
(c)

(g_1)				
0	8	0	0	0
1	0	9	0	0
2	0	0	10	0
3	0	0	0	9

→ (+1)
(a=1, b=1)
(d)

(g_1)				
0	0	0	9	0
1	0	0	0	10
2	9	0	0	0
3	0	8	0	0

→ (+1)
(a=1, b=2)
(e)

图 12-21 灰度共生矩阵计算方法

1）角二阶距

角二阶(ASM)距是图像灰度均匀性的一种度量，定义为

$$Q_1 = \sum_{g_1}\sum_{g_2}[p(g_1,g_2)]^2 \tag{12-73}$$

角二阶距是灰度共生矩阵元素值的平方和，因此又称为能量，它是纹理图像灰度分布均一性的度量，也被视为图像纹理粗细的度量指标。

如果共生矩阵的所有元素值均相等，则 ASM 值较小；反之，如果矩阵元素的一部分元素值大而其他值小，即共生矩阵中元素集中分布时，则 ASM 值大。

2）惯性矩

惯性矩又称为对比度(CON)，可以理解为图像的清晰度，在图像中纹理的纹沟越深，对比度则越大，图像的视觉效果越清晰。

$$Q_2 = \sum_{k}k^2\left[\sum_{g_1}\sum_{g_2}p(g_1,g_2)\right] \tag{12-74}$$

式中

$$k = g_1 - g_2 \tag{12-75}$$

对比度是图像中局部灰度变化大小的度量，对比度大小反映了图像的清晰度、纹理密度和沟纹深浅程度。图像的纹理沟纹越深，其对比度越大，视觉效果越清晰；反之，若对比度小，则沟纹浅，效果模糊。灰度差即对比度大的像素对越多，这个值越大。灰度共生矩阵中远离对角线的元素值越大，则对比度越大。

3）相关性

相关性(COR)用于衡量灰度共生矩阵的元素在行或列方向上的相似程度。

$$Q_3 = \frac{\sum_{g_1}\sum_{g_2}g_1g_2p(g_1,g_2) - \mu_x\mu_y}{\sigma_x\sigma_y} \tag{12-76}$$

式中

$$\begin{cases}\mu_x = \sum_{g_1} g_1 \sum_{g_2} p(g_1, g_2) \\ \mu_y = \sum_{g_2} g_2 \sum_{g_1} p(g_1, g_2)\end{cases} \tag{12-77}$$

$$\begin{cases}\sigma_x^2 = \sum_{g_1} (g_1 - \mu_x)^2 \sum_{g_1} \sum_{g_2} p(g_1, g_2) \\ \sigma_y^2 = \sum_{g_2} (g_2 - \mu_y)^2 \sum_{g_1} \sum_{g_2} p(g_1, g_2)\end{cases} \tag{12-78}$$

相关性是纹理图像局部灰度线性相关的度量，体现在纹理上就是表示图像中纹理区域在某种方向上的相似性。它用于度量空间灰度共生矩阵元素在行或列方向上的相似度，因此，相关性的大小反映了图像中局部灰度相关性。当元素均匀一致或相等时，相关值就大；反之，若矩阵元素值相差较大，则相关性小。图像内存在水平方向纹理时，则水平方向矩阵的 Q_3 值大于其余矩阵的 Q_3。

4）熵

熵(ENT)是图像具有的信息量的一种度量，纹理信息也是图像的信息，若一幅数字图像没有纹理，则灰度共生矩阵接近为零矩阵。

$$Q_4 = -\sum_{g_1} \sum_{g_2} p(g_1, g_2) \lg p(g_1, g_2) \tag{12-79}$$

熵表征了图像中纹理的复杂程度，反映了图像纹理灰度分布的随机性，是图像所具有的信息量的度量。纹理信息也属于图像的特征信息，当共生矩阵中所有元素有最大的随机性、空间共生矩阵中所有值几乎相等时，共生矩阵中元素分散分布时，熵较大。它表示了图像中纹理的非均匀程度或复杂程度。

5）逆差矩

逆差矩(IDM)的定义为

$$Q_5 = -\sum_{g_1} \sum_{g_2} p(g_1, g_2) [1 + (g_1 - g_2)^2] \tag{12-80}$$

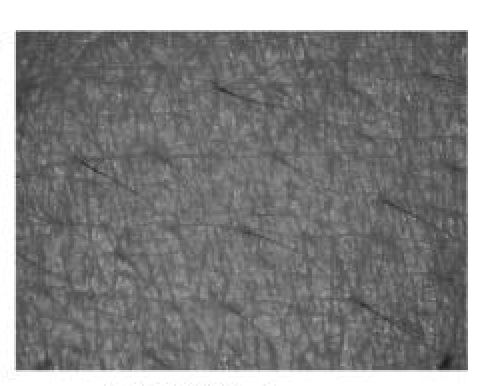
逆差矩反映了图像纹理的同质性，它可用于度量图像纹理的局部变化情况。逆差矩大则说明图像纹理的不同区域间缺少变化，局部非常均匀。

需指出的是，有些情况下，Q_1、Q_2、Q_3、Q_4、Q_5 单一指标所指示的图像特征并不是很直观，有时需要根据多个指标评价图像的纹理信息。根据目前对纹理的研究，$Q_1 \sim Q_5$ 是描述纹理特征非常有效的参数。

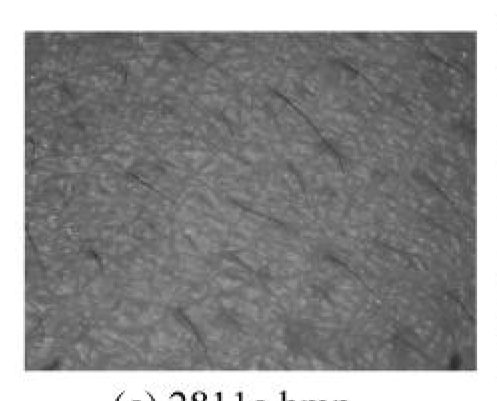
例 12-2：纹理特征参数计算

图 12-22 为 4 位不同个体的 4 幅面部脸颊皮肤图像，分别对上述图像采用基于灰度共生矩阵方法计算其纹理特征参数，包括角二阶矩、熵、对比度、相关，各所列特征值均为 0°、45°、90°、135° 4 个方向上的平均值，其标准差指特征值在上述 4 方向上的标准差。相应的实验数据如表 12-1 所示。

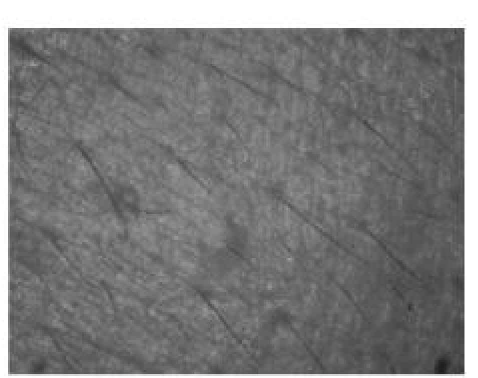
(a) 1801c.bmp



(b) 1802c.bmp



(c) 2811c.bmp



(d) 2812c.bmp

图 12-22　面部图像纹理图像

表 12-1　纹理特征参数计算结果

图像名	角二阶矩	标准差	熵	标准差	对比度	标准差	相关性	标准差
1801c	0.298 82	0.011 21	1.622 86	0.180 31	0.234 11	0.010 26	1.678 84	0.120 71
1802c	0.263 05	0.011 53	1.755 31	0.161 59	0.279 14	0.021 01	1.431 09	0.125 61
2811c	0.332 59	0.014 26	1.539 91	0.134 61	0.270 54	0.014 91	1.686 11	0.112 89
2812c	0.290 09	0.013 91	1.634 69	0.159 78	0.223 59	0.061 25	1.596 18	0.151 78

根据表 12-1 的实验数据，对于皮肤图像，其角二阶矩(ASM)大表明图像相对均一、规则变化的纹理模式，即图像的纹理结构较均匀一致。熵(ENT)越大说明图像的信息越大，即图像纹理变化较多；反之，则图像纹理变化较少。对比度(CON)大表明图像纹理结构差异性较大。相关性(COR)的大小表明皮肤图像中局部像元灰度值的相关性，即在行或列方向 GLCM 各元素的相似程度，COR 大说明皮肤图像的纹理相关性较好，纹理视觉效果较好。在进行大量皮肤纹理比较时，若皮肤图像的平均特征值相同，标准差小说明该特征值在 4 个方向上一致性更好。

本实验表明，纹理特征参数不仅能评价一幅图像纹理的信息，而且 4 个特征参数还具有相关性。若皮肤图像的熵值大，则对比度大，角二阶矩和相关性则小。

皮肤纹理特征参数实验程序代码如下：

```
[name,path] = uigetfile('D:\cth01\人脸纹理\*.bmp','打开皮肤图像文件');
name = strcat(path,name);
%fid = fopen(name);
Gray = imread(name);
[M,N,O] = size(Gray);
M = 256;
N = 256;
for i = 1:M
   for j = 1:N
      for n = 1:256/16
         if (n-1)*16 <= Gray(i,j)&Gray(i,j)<= (n-1)*16+15
            Gray(i,j) = n-1;
         end
      end
   end
end
%3.计算四个方向灰度共生矩阵,角度分别为0°,45°,90°,135°方向
P = zeros(16,16,4);
for m = 1:16
```

```
    for n = 1:16
      for i = 1:M
        for j = 1:N
          if j < N&Gray(i,j) == m - 1&Gray(i,j + 1) == n - 1
            P(m,n,1) = P(m,n,1) + 1;
            P(n,m,1) = P(m,n,1);
          end
          if i > 1&j < N&Gray(i,j) == m - 1&Gray(i - 1,j + 1) == n - 1
            P(m,n,2) = P(m,n,2) + 1;
            P(n,m,2) = P(m,n,2);
          end
          if i < M&Gray(i,j) == m - 1&Gray(i + 1,j) == n - 1
            P(m,n,3) = P(m,n,3) + 1;
            P(n,m,3) = P(m,n,3);
          end
          if i < M&j < N&Gray(i,j) == m - 1&Gray(i + 1,j + 1) == n - 1
            P(m,n,4) = P(m,n,4) + 1;
            P(n,m,4) = P(m,n,4);
         end
       end
     end
     if m == n
       P(m,n,:) = P(m,n,:) * 2;
     end
   end
end
% 共生矩阵归一化
for n = 1:4
   P(:,:,n) = P(:,:,n)/sum(sum(P(:,:,n)));
end
H = zeros(1,4);
I = H;
Ux = H;       Uy = H;
deltaX = H;  deltaY = H;
C = H;
for n = 1:4
   E(n) = sum(sum(P(:,:,n).^2)); % 能量
   for i = 1:16
      for j = 1:16
         if P(i,j,n)~ = 0
           H(n) = - P(i,j,n) * log(P(i,j,n)) + H(n); % 熵
         end
         I(n) = (i - j)^2 * P(i,j,n) + I(n); % 惯性矩
         Ux(n) = i * P(i,j,n) + Ux(n); % 相关性中 μx
         Uy(n) = j * P(i,j,n) + Uy(n); % 相关性中 μy
      end
   end
end
for n = 1:4
   for i = 1:16
      for j = 1:16
```

```
            deltaX(n) = (i - Ux(n))^2 * P(i,j,n) + deltaX(n);
            deltaY(n) = (j - Uy(n))^2 * P(i,j,n) + deltaY(n);
            C(n) = i * j * P(i,j,n) + C(n);
        end
    end
    C(n) = (C(n) - Ux(n) * Uy(n))/deltaX(n)/deltaY(n); % 相关性
end
% 能量、熵、惯性矩、相关均值和标准差
a1 = mean(E);
b1 = sqrt(cov(E));
a2 = mean(H) ;
b2 = sqrt(cov(H));
a3 = mean(I);
b3 = sqrt(cov(I));
a4 = mean(C);
b4 = sqrt(cov(C));
```

12.5.6 纹理句法结构分析

在纹理的句法结构分析中，将纹理定义为结构基元按某种规则重复分布所构成的模式。进行纹理结构分析，需要先描述结构基元的分布规律。因此，一般可进行如下两项工作：

(1) 从输入图像中提取结构基元并描述其特征。

(2) 描述结构基元的分布规则。具体方法如下：首先将纹理图像分成许多窗口，即形成子纹理。其中最小的小块就是最基本的子纹理(即基元)。纹理基元可以是一个像素，也可以是 4 个或 9 个灰度比较一致的像素集合。纹理的表达可以是多层次的，如图 12-23(a)所示，它可以从像素或小块纹理一层一层地向上拼合。而且，基元的排列可有不同规则，如图 12-23(b)中，第一级纹理排列为 ABA，第二级纹理排列为 BAB 等，其中 A、B 代表基元或子纹理，于是就组成了一个多层的树状结构，可用树状文法产生一定的纹理，并以句法加以描述。

纹理的树状安排可有多种方法。如图 12-23(c)所示，树根安排在中间，树枝向两边伸出，每个树枝有一定的长度。又如图 12-23(d)中，树根安排在一侧，分枝都向另一侧伸展。

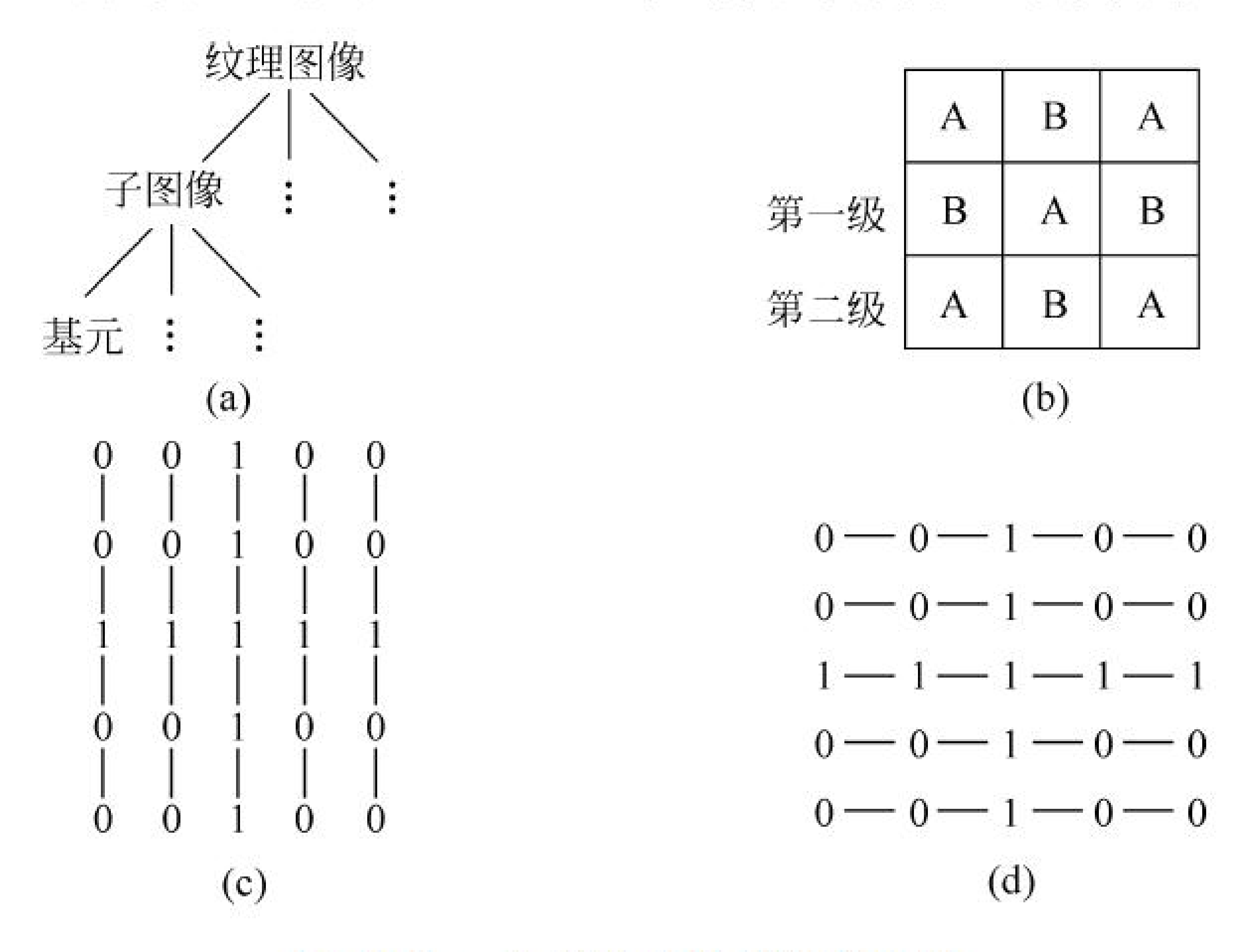


图 12-23 纹理的树状描述及排列

纹理判别可用以下方法：首先将纹理图像分成固定尺寸的窗口，用树状文法说明属于同纹理图像的窗口，可以用树状自动机识别树状，因此，对每一个纹理文法可建立一个"结构保存的误差修正树状自动机"。该自动机不仅可以接收每个纹理图像中的树，而且能用最小距离判据辨识类似的有噪声的树；然后，可以对一个分割成窗口的输入图像进行分类。

12.6 骨架提取

图像骨架提取是指提取目标在图像中的中心像素轮廓，即以物体的对称中心为目标，对目标进行细化，细化后的目标通常为单层像素宽度。图像细化和骨架提取的目的都是为了减小像素的连通区，一般情况下这些运算是不可逆的。通俗地说，骨架化运算是指将目标区域变为条形骨架的算法。由于从一幅骨架状图形常可以识别出一些复杂的形状(如动物、飞机等)，因此区域的骨架携带了物体的许多形状信息，它对描述变宽度的长条形物体且不需考虑其宽度影响的二值图像十分有效。

12.6.1 骨架的概念

目前，图像的骨架还没有被一致认可的定义。骨架或细化的概念主要是针对二值图而言的。某种意义上，所谓骨架，可以理解为图像的中轴，例如，长方形的骨架，主要由其长轴方向的中轴线组成，圆的骨架是其圆心，直线的骨架是它自身，孤立点的骨架也是其自身。

骨架的获取主要有以下两种方法。

1. 烈火燃烧

设想在同一时刻，将目标物体的边缘线同时点燃，全部燃烧点以匀速向内部蔓延，当火焰相交时熄灭，火焰熄灭点的集合就是目标的骨架。

2. 基于最大圆盘

根据几何学原理，目标的骨架是由目标内所有内切圆盘的圆心组成，如图 12-24 所示。

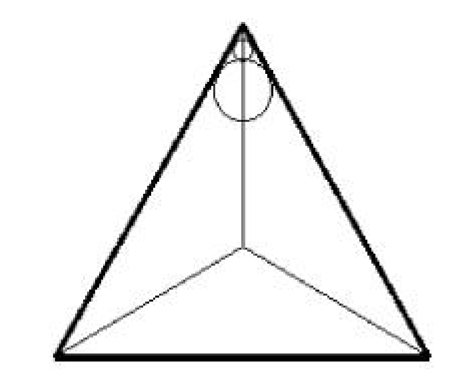


图 12-24 三角形形状物体的骨架示意图

12.6.2 中轴变换

将一个平面区域简化成图是一种重要的结构形状表示法。利用细化技术得到区域的骨架是常用的方法。中轴变换(medial axis transform，MAT)是一种用来确定物体骨架的细化技术。

设区域 A 的边界点集为 B，区域 A 的中轴变换算法过程是：对 A 中的每一点 P，在 B 中搜寻与它相距最近的点；如果区域内某一点有一个以上边界点与其有相同的最小距离，则点 P 必位于对称轴或中轴上。也就是说，B 中具有两个或两个以上的点与 P 同时最近，就可认为 P 属于 A 的中线或骨架，即点 P 是 A 的一个骨架点。中轴是离图像边界上两点具有最小相等距离的点集组成的，将按以上方法找出的中轴点连接起来，便得到中轴骨架。

理论上，由图形的中轴骨架及最小距离还可以重建图形的边界。由于每个骨架点保持了其与边界点距离最小的性质，因此，用以每个骨架点为中心的圆的集合(利用合适的量度)，就可恢复出原始的区域来。具体即以每个骨架点为圆心，以前述最小距离为半径作圆，

它们的包络就构成了区域的边界,填充这些圆就得到原区域。或者以每个骨架点为圆心,以所有小于和等于最小距离的长度为半径作圆,这些圆的并集就覆盖了整个区域。中轴变换原理如图 12-25 所示。

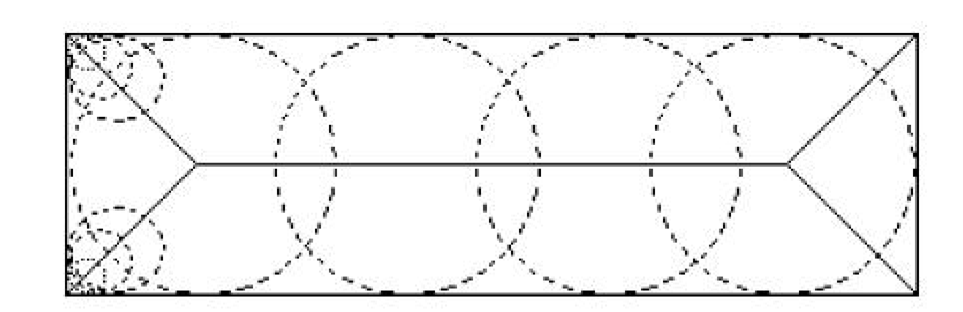


图 12-25 中轴变换原理示意图

根据上述讨论可知,骨架是用一个点与一个点集的最小距离来定义的,可写成

$$d_s(p,B) = \inf\{d(p,z \mid z \in B\} \tag{12-81}$$

上述定义中距离量度既可以是欧几里德距离,也可以是市区距离或棋盘距离。因为最小距离取决于所采用的距离量度,所以中轴变换的结果也和所采用的距离量度有关。骨架的提取也可以采用数学形态学方法。在提取出骨架后,很容易根据原图计算出每点到边界的最短距离。

图 12-26 给出了 3 种类型的区域及采用欧氏距离测度算出的骨架。根据图 12-26(a)可知,较粗短的物体骨架提供的信息则较少;图 12-26(b)和(c)则表明,较细长区域或物体的骨架通常可提供较多的形状信息。

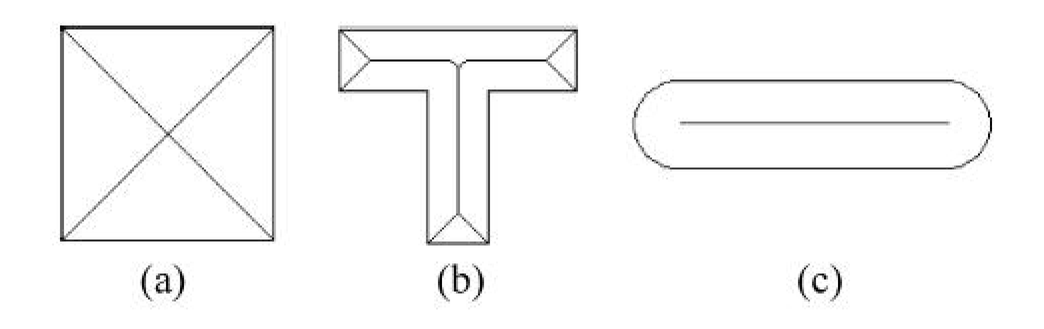


图 12-26 几种类型的区域和物体欧氏距离骨架

在某些情况下,采用骨架表示区域可能受噪声的影响较大。例如,图 12-27(b)中的区域与图 12-27 (a)中的区域略有差别,可认为图 12-27(a)由于噪声的影响成为了图 12-27(b),图 12-27(a)和图 12-27(b)两者轮廓差别甚小,但两者的骨架相差很大。

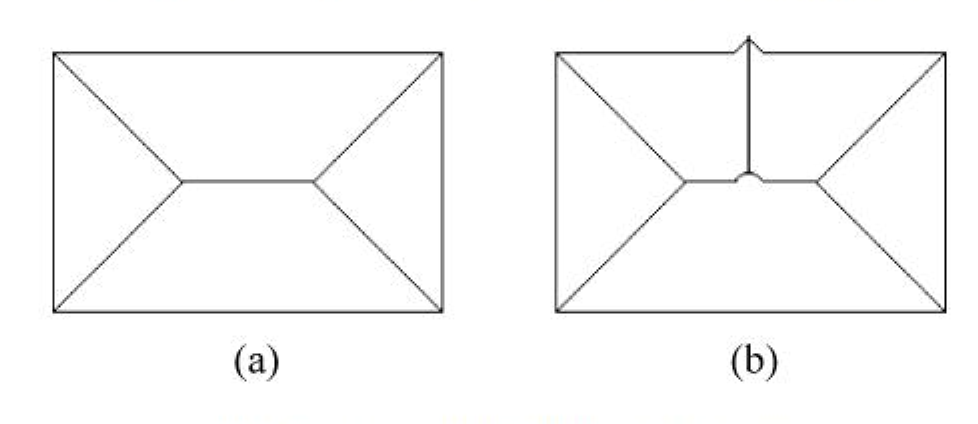


图 12-27 骨架受噪声的影响

一般情况下,根据式(12-81)求区域骨架需要计算所有边界点到所有区域内部点的距离,计算量较大。实际应用中,求区域骨架一般采用逐次消去边界点的迭代细化算法。应用该方法需要注意如下原则:

(1) 不能消除线段的端点。

(2) 不中断原来连通的点。

(3) 不过多侵蚀区域。

由上述分析可以看出，图像骨架提取算法实际上是依据一定的代数、几何等数学原理，提取目标在图像上的中心像素轮廓。基于数学原理的骨架提取算法多种多样，图像的骨架一般由一些线和曲线（理想情况是单像素宽度）组成。图像的骨架可以提供一个目标的尺寸和形状信息，在数字图像分析中具有重要的地位，图像骨架化是进行图像识别、线性图像目标分析的重要手段。

习题

1. 图像的几何特征包括哪些方面？各有何用途？

2. 试简述如何检测一个区域或物体的长轴和短轴。

3. 数字图像分析中计算区域的周长有几种方法？各有何特点？

4. 选择几个常用的机械零件或者五金工具图形，编写程序测量这些图像的几何特征参数。

5. 试根据图 12-28 所示图像计算图中阴影区域的周长。

6. 数字图像处理技术中，计算区域的面积主要有哪些方法？

7. 数字图像的距离有几种计算方法？试写出这些计算公式。

8. 什么是物体的形状特征？它主要包括哪些方面？

9. 彩色图像的颜色特征有哪些？试分别阐述什么是彩色图像的颜色矩和颜色聚合向量。

10. 什么是傅里叶描述子？傅里叶描述子具有哪些特点？

11. 选择一个轮廓复杂的物体和轮廓相对简单的物体，按要求完成以下实验：

(1) 编写程序分别对其轮廓形状进行傅里叶描述。

(2) 将两轮廓分别进行一定角度的旋转后再进行傅里叶轮廓描述。

(3) 将旋转前后的形状描述结果进行分析对照，总结傅里叶描述子的特点。

12. 试根据如图 12-29 所示图像，分别计算图中阴影区域的 4 方向和 8 方向边界链码。

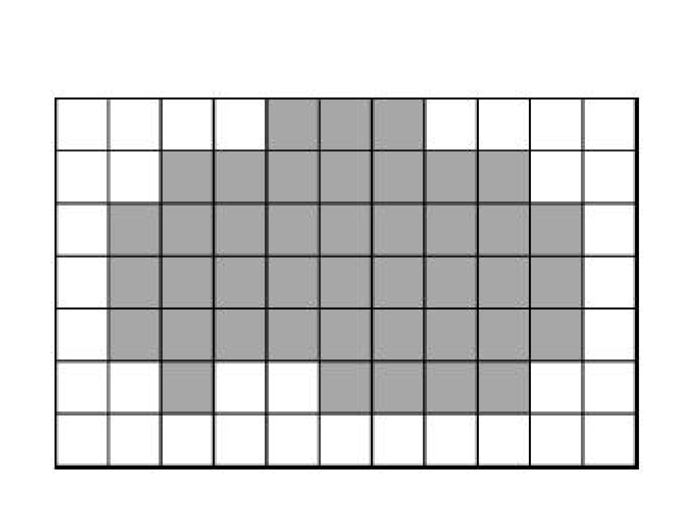


图 12-28 物体的周长计算

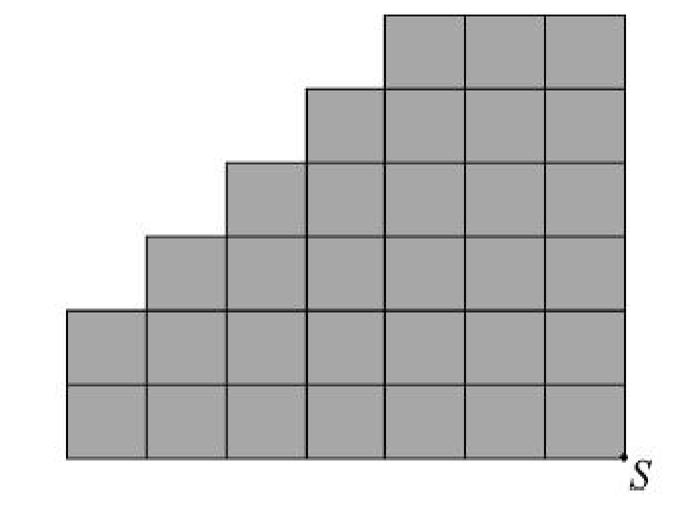


图 12-29 区域的边界链码

13. 试根据霍夫变换的原理，证明变换域极坐标系中的一点对应于直角坐标系中的一条直线。

14. 选择一张带表格的票据图像，检测表格的倾斜角度，并进行校正。

15. 什么是纹理分析中的统计方法？它有何特点？

16. 简述计算灰度共生矩阵的主要步骤，编写程序实现灰度共生矩阵算法。

17. 试简述联合概率矩阵法的基本原理。

18. 选择若干不同人员同一部位的皮肤图像，编写程序计算各图像的角二阶距、对比度、相关性、熵等粗糙度指标。

(1) 简述各项指标的物理意义。

(2) 依据计算所得出的四项指标，分别按大小进行排序。

(3) 按照排序，逐一分析皮肤指标与视觉感受是否一致。

19. 试简述中轴变换的主要原理。

20. 简要阐述骨架提取算法在图像处理中的作用和意义，并编写程序对图像进行骨架提取。

参 考 文 献

[1] 章毓晋.图像工程[M].4版.北京：清华大学出版社，2018.

[2] 拉斐尔·C.冈萨雷斯，理查德·E.伍兹.数字图像处理[M].3版.阮秋琦，阮宇智，等译.北京：电子工业出版社，2017.

[3] 朱虹.数字图像处理基础[M].北京：科学出版社，2017.

[4] 卡斯尔曼.数字图像处理[M].朱志刚，林学訚，石定机，等译，北京：电子工业出版社，2011.

[5] 阮秋琦.数字图像处理学[M].3版.北京：电子工业出版社，2013.

[6] 姚敏.数字图像处理[M].3版.北京：机械工业出版社，2017.

[7] 赵银娣.遥感数字图像处理[M].北京：测绘出版社，2015.

[8] 夏良正.数字图像处理[M].2版.南京：东南大学出版社，2011.

[9] 张铮，倪红霞，苑春苗，等.精通MATLAB数字图像处理与识别[M].北京：人民邮电出版社，2013.

[10] 赵荣椿，赵忠明，赵歆波.数字图像处理与分析[M].北京：清华大学出版社，2013.

[11] 蒋爱平，王晓飞，杜宝祥.数字图像处理[M].北京：科学出版社，2013.

[12] 何斌，马天予.VC数字图像处理[M].北京：人民邮电出版社，2010.

[13] 胡学龙.数字图像处理[M].3版.北京：电子工业出版社，2014.

[14] 李俊山，李旭辉，朱子江.数字图像处理[M].北京：清华大学出版社，2017.

[15] 何东健.数字图像处理[M].西安：西安电子科技大学出版社，2012.

[16] 朱秀昌，刘峰，胡栋.数字图像处理与图像通信[M].北京：北京邮电大学出版社，2008.

[17] 韩晓军.数字图像处理技术与应用[M].北京：电子工业出版社，2017.

[18] 杨帆，丁士心，唐红梅.数字图像处理与分析[M].2版.北京：北京航空航天大学出版社，2013.

[19] 莫德举.数字图像处理[M].北京：北京邮电大学出版社，2010.

[20] 陈莉.数字图像处理基础[M].北京：科学出版社，2017.

[21] 罗述谦，周果宏.医学图像处理与分析[M].北京：科学出版社，2010.

[22] 曹茂永.数字图像处理[M].北京：北京大学出版社，2013.

[23] 郭文强.数字图像处理[M].西安：西安电子科技大学出版社，2012.

[24] 程光权.基于几何特征的图像处理与质量评价[M].北京：国防工业出版社，2013.

[25] 霍红涛.数字图像处理[M].北京：北京理工大学出版社，2006.

[26] 侯宏花.数字图像处理与分析[M].北京：北京理工大学出版社，2011.

[27] 全红艳，曹桂涛.数字图像处理原理与实现方法[M].北京：机械工业出版社，2014.

[28] 程正兴.小波分析算法与应用[M].西安：西安交通大学出版社，2012.

[29] 徐超.数字图像处理与机器视觉——Visual C++与MATLAB实现[M].北京：人民邮电出版社，2016.